Axiomatisation and Decidability of Multi-Dimensional Duration Calculus

Andreas Schäfer
Department of Computing Science,
University of Oldenburg, 26111 Oldenburg, Germany
schaefer@informatik.uni-oldenburg.de

Abstract

We investigate properties of a spatio-temporal logic based on an n-dimensional Duration Calculus tailored for the specification and verification of mobile real-time systems. After showing non-axiomatisability, we give a complete embedding in n-dimensional interval temporal logic and present two different decidable subsets, which are important for tool support and practical use.

Keywords: *Real-time systems, mobile systems, spatial logic, temporal logic, Duration Calculus*

1. Introduction

Real-time systems are used in many areas of our life and their use even increases further, for example in embedded systems for cars, train- and traffic controllers. There are numerous well studied formal methods for describing and reasoning about such systems, like Duration Calculus (DC) [15] or timed automata [1]. But the behaviour of many real-time systems exhibits mobile aspects. Railroad tracks are divided into blocks and it is required that for every moment in time each block is occupied by at most one train. Airplanes are required to have a minimum distance during their flight. Or for a mobile robot it might be important, that it does not enter a certain area by more than 10 cm.

As Duration Calculus has already proven its beneficial applicability in the area of real-time systems, an n-dimensional extension, the shape calculus (SC), is proposed to grasp this quantitative notion of space and mobility. In this paper we investigate properties of this formalism like axiomatisability and decidability. These points are crucial as acceptance in practice often requires tool support by model-checkers or theorem provers.

After giving a short introduction to SC in Section 2, we show that SC is not axiomatisable, but nevertheless it can be completely axiomatised relative to the n-dimensional extension of interval temporal logic. In Section 4 we present two decidable subsets of discrete SC, one achieved by imposing restrictions on the class of models, one by restriction on the class of formulae.

2. Shape Calculus

In this section we introduce the shape calculus proposed in [13] in a simplified version.

In Duration Calculus [15] the behaviour of a system is modelled by a set of time-dependent variables (observables) whose values change in time. Here we adopt this approach and use boolean observables that are space and time dependent. We may choose to have discrete or continuous time and space depending on the current application. A priori we fix the number of spatial and temporal dimensions, say n then the semantics of an observable X is given by a trajectory $\mathcal{I}$

$$\mathcal{I}[\![X]\!] : \mathbb{R}^n_{\geq 0} \to \{0, 1\}.$$

in the continuous case or a function with domain $\mathbb{N}^n$ in the discrete case.

Example 1 *To model a mobile robot moving on the floor, we need two spatial and one temporal dimension, so we fix $n = 3$. We employ two observables R and A. The observable R is true for a point in space and time if and only if the robot occupies this point in space at the given moment in time. Similarly the restricted area is modeled by the observable A.*

As we will measure time and space, we have to guarantee that an integral exists. So we require a finite variability property, that is every finite n-dimensional interval can be partitioned in finitely many sub-intervals such that $\mathcal{I}$ is constant on each sub-interval.

The language of SC is built from state expressions, terms and formulae. A state expression characterises a property of one point in time and space. They are denoted by π and built from boolean combinations of observables. The semantics is a function $\mathcal{I}[\![\pi]\!] : \mathbb{R}^n_{\geq 0} \rightarrow \{0,1\}$ and is defined as a straightforward extension of trajectories of observables.

$$\mathcal{I}[\![\neg\pi]\!](\vec{z}) \stackrel{df}{=} 1 - \mathcal{I}[\![\pi]\!](\vec{z})$$
$$\mathcal{I}[\![\pi \wedge \pi']\!](\vec{z}) \stackrel{df}{=} \mathcal{I}[\![\pi]\!](\vec{z}) \cdot \mathcal{I}[\![\pi']\!](\vec{z})$$

Example 2 *The state expression $R \wedge \neg A$ describes exactly the points in space-time where the robot is outside its restricted area.*

A term θ is either a measure $\int \pi$, a rigid variable x, the special symbol $\ell_{\vec{e}_i}$ denoting the length in direction along $\vec{e}_i$ or the application of a function f. Usually we will use functions like summation or multiplication.

$$\theta ::= \int \pi \mid x \mid \ell_{\vec{e}_i} \mid f(\theta_1, \ldots, \theta_k)$$

As usual, the value of a rigid variable is determined by a valuation $\mathcal{V}$ and the set of valuations is denoted by Val. The semantics assigns a real number to each n-dimensional interval and thus it is a function $\mathcal{I}[\![\theta]\!] : \mathsf{Int}^n \times Val \rightarrow \mathbb{R}$ and defined in the expected way, i.e. let

$$\mathcal{M} = [b_1, e_1] \times \ldots [b_i, e_i] \ldots \times [b_n, e_n] \in \mathsf{Int}^n$$

and $\mathcal{V} \in Val$ then

$$\mathcal{I}[\![\textstyle\int \pi]\!](\mathcal{M}, \mathcal{V}) \stackrel{df}{=} \int_{\mathcal{M}} \mathcal{I}[\![\pi]\!]$$
$$\mathcal{I}[\![\ell_{\vec{e}_i}]\!](\mathcal{M}, \mathcal{V}) \stackrel{df}{=} e_i - b_i$$
$$\mathcal{I}[\![x]\!](\mathcal{M}, \mathcal{V}) \stackrel{df}{=} \mathcal{V}(x)$$
$$\mathcal{I}[\![f(\theta_1, \ldots, \theta_k)]\!](\mathcal{M}, \mathcal{V}) \stackrel{df}{=} f_{\mathcal{I}}(\ \mathcal{I}[\![\theta_1]\!](\mathcal{M}, \mathcal{V}) \ \ldots \ \mathcal{I}[\![\theta_k]\!](\mathcal{M}, \mathcal{V}))$$

Example 3 *The term $\int R \wedge \neg A$ is the measure of all points violating the requirement.*

Formulae are interpreted over intervals and as usual for interval logics, they incorporate a special "chop" operator to partition the current interval into two parts. As we consider a higher dimensional logic, we allow chops along each axis. Formally we define the set of *formulae* by

$$F ::= F_1 \langle \vec{e_i} \rangle F_2 \mid p(\theta_1, \ldots, \theta_k) \mid \neg F_1 \mid F_1 \wedge F_2 \mid \exists x :\ F$$

where p is a predicate symbol like $=$ oder $\leq$, x a rigid variable and $\vec{e_i}$ the i-th unit vector. The other boolean connectives can be defined as the usual abbreviations. We only need to give the definition of "chop" here as the other operators and the existential quantifier are defined as usual.

$$\mathcal{I}[\![F_1 \langle \vec{e_i} \rangle F_2]\!](\mathcal{M}, \mathcal{V}) = \text{true}$$

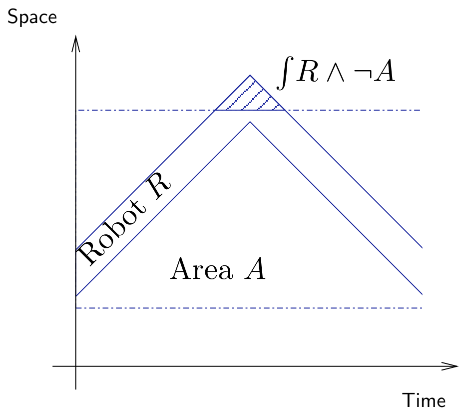

Figure 1. Moving robot scenario

iff $\mathcal{M} = [b_1, e_1] \times \ldots \times [b_i, e_i] \times \ldots \times [b_n, e_n]$ and there is a $m \in [b_i, e_i]$ such that

$$\mathcal{I}[\![F_1]\!]([b_1, e_1] \times \ldots [b_i, m] \ldots \times [b_n, e_n], \mathcal{V}) = \text{true and}$$
$$\mathcal{I}[\![F_2]\!]([b_1, e_1] \times \ldots [m, e_i] \ldots \times [b_n, e_n], \mathcal{V}) = \text{true.}$$

We define some abbreviations to ease the handling. The everywhere operator $\lceil \pi \rceil$ expresses that a state assertion π holds almost everywhere in the interval and the empty interval is denoted by $\lceil\rceil$. The n-dimensional volume is measured by the term ℓ.

$$\ell \stackrel{df}{=} \int 1 \qquad \lceil \pi \rceil \stackrel{df}{=} \int \pi = \ell \wedge \ell > 0 \qquad \lceil\rceil \stackrel{df}{=} \int 1 = 0$$

The somewhere operator $\Diamond_{\vec{e}_i} F$ chops the n-dimensional interval twice in the same direction such that in the middle interval F holds.

$$\Diamond_{\vec{e}_i} F \stackrel{df}{=} \text{true} \langle \vec{e_i} \rangle\ F\ \langle \vec{e_i} \rangle \text{ true}$$

The dual globally operator is $\Box_{\vec{e}_i}$ defined by

$$\Box_{\vec{e}_i} \stackrel{df}{=} \neg \Diamond_{\vec{e}_i} \neg F.$$

We will call the unit vector corresponding to the time dimension $\vec{e}_t$ and to spatial dimensions $\vec{e}_x, \vec{e}_y, \ldots$.

Example 4 *The initial requirement, that a robot R does never enter a restricted area A by more than 10 cm can be expressed by*

$$\Box_{\vec{e}_t} \textstyle\int (R \wedge A) \leq (10 \cdot \ell_{\vec{e}_t}).$$

The scenario is sketched in figure 1. The observable R modelling the robot is true for all points between the solid lines, the observable A is true for all points between the dashed lines. For simplicity we omitted the second spatial dimension in the drawing.

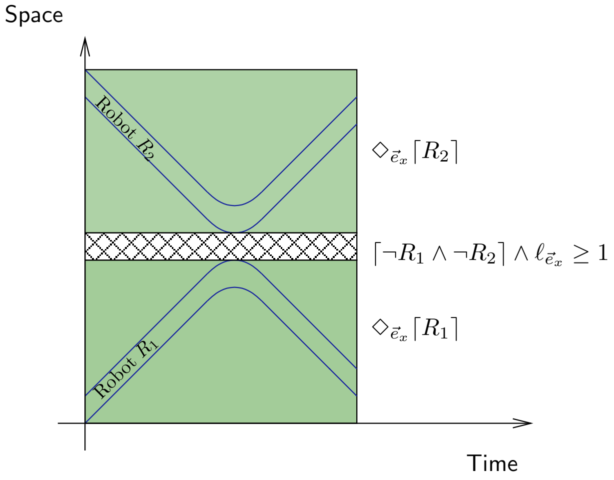

Figure 2. Minimal distance scenario

Example 5 (Ensuring a minimal distance) *Consider the scenario of two moving robots using a collision avoidance system as depicted in figure 2. We require that the minimal distance is always greater than $1m$. This is specified by*

$$\Box_{\vec{e}_t}\Box_{\vec{e}_x}$$
$$\big((\Diamond_{\vec{e}_x}\lceil R_1\rceil \wedge \Diamond_{\vec{e}_x}\lceil R_2\rceil)$$
$$\Rightarrow$$
$$(\Diamond_{\vec{e}_x}\lceil R_1\rceil \langle\vec{e}_x\rangle (\lceil\neg R_1 \wedge \neg R_2\rceil \wedge \ell_{\vec{e}_x} \geq 1) \langle\vec{e}_x\rangle \Diamond_{\vec{e}_x}\lceil R_2\rceil)\big)$$

This formula reads as follows. For all spatio-temporal subintervals such that Robot R_1 and Robot R_2 are contained somewhere in this interval we can split space into three parts such that

- *the lower part contains R_1*
- *the middle part does neither contain Robot R_1 nor R_2 and has length greater or equal 1*
- *and the upper part contains R_2.*

Definition 1 (Validity / Satisfiability) *A formula F is called valid iff it holds for all interpretations, valuations and intervals. It is satisfiable iff there is an interpreation, a valuation and an interval that makes F hold.*

3. Axiomatisability

In [8] it is shown that discrete time Duration Calculus is decidable. But this does no longer hold for more than one dimension.

Theorem 1 *For two dimensions and above SC is not recursively enumerable, neither interpreted in the continuous nor in the discrete domain.*

Corollary 1 *There is no sound and complete proof system for SC.*

The proof is similar to the undecidablity result in [13]. It is done by reduction from the emptiness problem for tiling systems to the satisfiability problem for SC.

For this proof we restrict ourselves to the set of formulae given by

$$F ::= \lceil\pi\rceil \mid F \wedge G \mid \neg F \mid F\langle\vec{e}_1\rangle G \mid F\langle\vec{e}_2\rangle G \mid \ell_{\vec{e}_i} = r$$

for some fixed r. Without loss of generality, we choose $r = 1$.

3.1. Tiling Systems

We fix an alphabet Σ and a special character $\#$. A tile p is a 2×2 matrix with elements in $\Sigma \cup \#$ and a tiling system Θ is a finite set of tiles. The local language $L(\Theta)$ for a tiling system Θ is the set of all $n\times m$ matrices such that each 2×2 block is in Θ and the boundaries of the matrix consist only of $\#$ and $\#$ does not occur in the interior.

Giammarresi and Restivo show in [6] that the emptiness problem

Given a tiling system Θ, is $L(\Theta) = \varnothing$?

is undecidable. They provide a reduction such that a Turing Machine $\mathcal{M}$ has no successful computation iff $L(\Theta)$ is empty. So the emptiness problem for tiling systems is not recursively enumerable.

3.2. Encoding tilings in shape calculus

For a set of tiles $\Theta = \{p_1, \ldots, p_k\}$ we define a formula F_Θ in SC, such that $L(\Theta) \neq \varnothing$ iff F_Θ is satisfiable. We present an encoding which does not rely on continuous or discrete time and space domain. Therefore we forbid chopping at arbitrary positions by imposing a chess-board marking by a fresh observable $\bigstar$ to clearly identify 2×2 blocks in the continuous case. We define for every tile p_i a formula F_{p_i} and then formalise in F_Θ that each 2×2 block is a valid tile. The grid is defined by

$$\begin{aligned}
F_{grid} \stackrel{df}{=} & ((\lceil\bigstar\rceil \wedge \ell_{\vec{e}_1} = 1 \wedge \ell_{\vec{e}_2} = 1 \langle\vec{e}_1\rangle \text{true}) \langle\vec{e}_2\rangle \text{true}) \\
& \wedge (\text{true} \langle\vec{e}_1\rangle (\text{true} \langle\vec{e}_2\rangle ((\lceil\bigstar\rceil \vee \lceil\neg\bigstar\rceil) \\
& \qquad \wedge \ell_{\vec{e}_1} = 1 \wedge \ell_{\vec{e}_2} = 1))) \\
& \wedge \neg\Diamond_{\vec{e}_1}\Diamond_{\vec{e}_2}(\lceil\bigstar\rceil \wedge (\ell_{\vec{e}_1} > 1 \vee \ell_{\vec{e}_2} > 1)) \\
& \wedge \neg\Diamond_{\vec{e}_1}\Diamond_{\vec{e}_2}(\lceil\neg\bigstar\rceil \wedge (\ell_{\vec{e}_1} > 1 \vee \ell_{\vec{e}_2} > 1)) \\
& \wedge \neg\Diamond_{\vec{e}_1}\Diamond_{\vec{e}_2}(\lceil\bigstar\rceil \langle\vec{e}_1\rangle (\lceil\neg\bigstar\rceil \wedge \ell_{\vec{e}_1} < 1) \langle\vec{e}_1\rangle \lceil\bigstar\rceil) \\
& \wedge \neg\Diamond_{\vec{e}_1}\Diamond_{\vec{e}_2}(\lceil\bigstar\rceil \langle\vec{e}_2\rangle (\lceil\neg\bigstar\rceil \wedge \ell_{\vec{e}_2} < 1) \langle\vec{e}_2\rangle \lceil\bigstar\rceil) \\
& \wedge \neg\Diamond_{\vec{e}_1}\Diamond_{\vec{e}_2}(\lceil\neg\bigstar\rceil \langle\vec{e}_1\rangle (\lceil\bigstar\rceil \wedge \ell_{\vec{e}_1} < 1) \langle\vec{e}_1\rangle \lceil\neg\bigstar\rceil) \\
& \wedge \neg\Diamond_{\vec{e}_1}\Diamond_{\vec{e}_2}(\lceil\neg\bigstar\rceil \langle\vec{e}_2\rangle (\lceil\bigstar\rceil \wedge \ell_{\vec{e}_2} < 1) \langle\vec{e}_2\rangle \lceil\neg\bigstar\rceil)
\end{aligned}$$

$\bigstar \wedge \#$	$\#$	$\bigstar \wedge \#$	$\#$	$\bigstar \wedge \#$
$\#$	$\bigstar \wedge a$	b	$\bigstar \wedge a$	$\#$
$\bigstar \wedge \#$	c	$\bigstar \wedge a$	b	$\bigstar \wedge \#$
$\#$	$\bigstar \wedge a$	b	$\bigstar \wedge a$	$\#$
$\bigstar \wedge \#$	$\#$	$\bigstar \wedge \#$	$\#$	$\bigstar \wedge \#$

Figure 3. Sample encoding of tilings in a grid structure

which requires that $\bigstar$ and $\neg\bigstar$ alternate in distance 1 starting with $\bigstar$. The first subformula requires a 1×1 $\bigstar$-block in the lower left corner, the second subformula specifies a full 1×1 $\bigstar$ or $\neg\bigstar$ block in the upper right corner. The other subformulae specify that each block is at least and at most 1×1. To describe a 2×2 block in this grid satisfying $P_1, \ldots P_4$ in its four cells we use the pattern

$$\begin{aligned} F_{2\times 2}(P_1, P_2, P_3, P_4) \stackrel{df}{=} & ((\lceil \bigstar \wedge P_1 \rceil \langle \vec{e_1} \rangle (\lceil \neg\bigstar \wedge P_2 \rceil) \langle \vec{e_2} \rangle \\ & (\lceil \neg\bigstar \wedge P_3 \rceil) \langle \vec{e_1} \rangle (\lceil \bigstar \wedge P_4 \rceil)) \vee \\ & ((\lceil \neg\bigstar \wedge P_1 \rceil) \langle \vec{e_1} \rangle (\lceil \bigstar \wedge P_2 \rceil) \langle \vec{e_2} \rangle \\ & (\lceil \bigstar \wedge P_3 \rceil) \langle \vec{e_1} \rangle (\lceil \neg\bigstar \wedge P_4 \rceil)) \end{aligned}$$

Now we can map every tile $p_i = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ to a formula $F_{p_i} \stackrel{df}{=} F_{2\times 2}(a, b, c, d)$. With these sub-formulae we define F_Θ to be

$$\begin{aligned} F_\Theta \stackrel{df}{=} & F_{grid} \\ & \wedge \Box_{\vec{e_1}} \Box_{\vec{e_2}} (F_{2\times 2}(\text{true}, \text{true}, \text{true}, \text{true}) \Rightarrow \bigvee_{i=1}^{k} F_{p_i}) \\ & \wedge \lceil \# \rceil \langle \vec{e_1} \rangle (\lceil \# \rceil \langle \vec{e_2} \rangle \lceil \neg\# \rceil \langle \vec{e_2} \rangle \lceil \# \rceil) \langle \vec{e_1} \rangle \lceil \# \rceil \\ & \wedge \lceil \bigwedge_{s, s' \in \Sigma, s \neq s'} s \Rightarrow \neg s' \rceil \end{aligned}$$

The second part describes, that each 2×2 block in the grid must be in Θ, whereas the third part defines that the picture must be framed by $\#$ and $\#$ does not occur in the interior, as sketched in figure 3. The last part ensures mutual exclusion of symbols. With this definition F_Θ is satisfiable if and only if the local language $L(\Theta)$ is not empty, so $\neg F_\Theta$ is valid if and only if the local language $L(\Theta)$ is empty.

So SC is not recursively enumerable and not axiomatisable. $\Box$

3.3. Relative Axiomatisation

Duration Calculus itself only allows an axiomatisation relative to interval temporal logic (ITL) [7]. Despite the negative result of the previous section, it is still possible to give an axiomatisation of SC relative to the n-dimensional variant of Interval Temporal Logic (ITL^n). Our proof follows the lines of [7, 8] and considers only the 2-dimensional case, but it can be easily generalised to more dimensions. ITL^n does not use state assertions or the integral operator and uses flexible variables v whose values depend on the interval, rigid variables x and $\ell_{\vec{e_i}}$ as terms.

$$\theta^{\mathsf{ITL}^n} ::= x \mid v \mid \ell_{\vec{e_i}} \mid f(\theta_1^{\mathsf{ITL}^n}, \ldots, \theta_k^{\mathsf{ITL}^n})$$

Furthermore we define the abbreviation $\ell \stackrel{df}{=} \ell_{\vec{e_1}} \cdot \ell_{\vec{e_2}}$ to measure the volume. For formulae it allows boolean combination, chop and quantification as in SC.

$$\begin{aligned} F^{\mathsf{ITL}^n} ::= & F_1^{\mathsf{ITL}^n} \langle \vec{e_i} \rangle F_2^{\mathsf{ITL}^n} \mid p(\theta_1^{\mathsf{ITL}^n}, \ldots, \theta_k^{\mathsf{ITL}^n}) \mid \\ & \neg F_1^{\mathsf{ITL}^n} \mid F_1^{\mathsf{ITL}^n} \wedge F_2^{\mathsf{ITL}^n} \mid \exists x : F^{\mathsf{ITL}^n} \end{aligned}$$

Theorem 2 *SC is then axiomatised relative to ITL^n by the following axioms.*

$$\textstyle\int 0 = 0 \qquad \text{(SC1)}$$
$$\textstyle\int 1 = \ell \qquad \text{(SC2)}$$
$$\textstyle\int \pi \geq 0 \qquad \text{(SC3)}$$
$$\textstyle\int \pi_1 + \int \pi_2 = \int (\pi_1 \vee \pi_2) + \int (\pi_1 \wedge \pi_2) \qquad \text{(SC4)}$$
$$\textstyle\int \pi = x \langle \vec{e_i} \rangle \int \pi = y \Rightarrow \int \pi = x + y \qquad \text{(SC5)}$$
$$\lceil\rceil \vee ((\lceil \pi \rceil \vee \lceil \neg\pi \rceil \langle \vec{e_1} \rangle\, true) \langle \vec{e_2} \rangle\, true) \qquad \text{(FV1)}$$
$$\lceil\rceil \vee ((\lceil \pi \rceil \vee \lceil \neg\pi \rceil \langle \vec{e_1} \rangle\, true) \langle -\vec{e_2} \rangle\, true) \qquad \text{(FV2)}$$
$$\lceil\rceil \vee ((\lceil \pi \rceil \vee \lceil \neg\pi \rceil \langle -\vec{e_1} \rangle\, true) \langle \vec{e_2} \rangle\, true) \qquad \text{(FV3)}$$
$$\lceil\rceil \vee ((\lceil \pi \rceil \vee \lceil \neg\pi \rceil \langle -\vec{e_1} \rangle\, true) \langle -\vec{e_2} \rangle\, true) \qquad \text{(FV4)}$$

The axioms (FV1)-(FV4) specify finite variability, by demanding that for every point we can find 4 rectangles to the lower left, lower right, upper left and upper right respectively such that the value of a state expression is constant. We introduce negated unit vectors and define $F \langle -\vec{e_i} \rangle G \stackrel{df}{=} G \langle \vec{e_i} \rangle F$ to make the presentation more concise.

Proof. The proof of relative completeness proceeds as follows. For a valid SC formula F we construct the ITL^n formula F^I by replacing the measure $\int \pi$ with a variable $v_{[\pi]}$. We use the superscript I to indicate an ITL^n formula obtained from an SC formula by this replacement. The needed instances of the SC axioms are encoded by an ITL^n formula H_F^I such that $\Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$ is valid. We then assume an ITL^n deduction $\vdash_{\mathsf{ITL}^n} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$. This deduction is an SC deduction $\vdash_{\mathsf{SC}} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F \Rightarrow F$. As $\Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F$ is a conjunction of SC axioms, we can easily deduce $\vdash_{\mathsf{SC}} F$ by modus ponens.

Let F be an arbitrary valid SC formula and let $X_1, \ldots X_l$ be the set of boolean observables occurring in F and $\mathcal{S}$ the set of all state expressions build from these observables. Only finitely many state expressions can be non equivalent. Let $[\pi] \stackrel{df}{=} \{\pi' \mid \pi \iff \pi' \text{ in propositional logic}\}$ denote such an equivalence class and $\mathcal{S}_{\equiv} = \{[\pi] | \pi \in \mathcal{S}\}$ the set of equivalence classes. For every equivalence class $[\pi]$ we introduce an ITL^n flexible variable $v_{[\pi]}$ with the intuition that $v_{[\pi]}$ models the duration $\int \pi$.

We encode the SC Axioms by the following finite sets of ITL^n formulae.

$$\mathcal{H}_1 \stackrel{df}{=} \{v_{[0]} = 0\}$$

$$\mathcal{H}_2 \stackrel{df}{=} \{v_{[1]} = \ell\}$$

$$\mathcal{H}_3 \stackrel{df}{=} \{v_{[\pi]} \geq 0 \mid [\pi] \in \mathcal{S}_{\equiv}\}$$

$$\mathcal{H}_4 \stackrel{df}{=} \{v_{[\pi_1]} + v_{[\pi_2]} = v_{[\pi_1 \vee \pi_2]} + v_{[\pi_1 \wedge \pi_2]} \mid [\pi_1], [\pi_2] \in \mathcal{S}_{\equiv}\}$$

$$\mathcal{H}_5 \stackrel{df}{=} \{v_{[\pi]} = x \langle \vec{e_i} \rangle v_{[\pi]} = y \Rightarrow v_{[\pi]} = x + y \mid [\pi] \in \mathcal{S}_{\equiv}\}$$

$$\mathcal{H}_6 \stackrel{df}{=} \{\lceil\rceil \vee ((\lceil v_{[\pi]} \rceil \vee \lceil v_{[\neg\pi]} \rceil \langle \vec{d_1} \rangle \, true) \langle \vec{d_2} \rangle \, true) \mid [\pi] \in \mathcal{S}_{\equiv}, d_i \in \{\vec{e}_i, -\vec{e}_i\}\}$$

where $\lceil v_{[\pi]} \rceil \stackrel{df}{=} v_{[\pi]} = \ell \wedge \ell > 0$ and $\lceil\rceil \stackrel{df}{=} \ell_1 = 0 \vee \ell_2 = 0$. We define H_F^I to be the conjunction of all formulae in $\mathcal{H}_1$ to $\mathcal{H}_6$ and F^I to be the ITL^n formula obtained from F by replacing every occurrence of $\int \pi$ by $v_{[\pi]}$.

Definition 2 (H-Triple) *A triple $(\mathcal{I}, \mathcal{V}, [b_1, e_1] \times [b_2, e_2])$ is called a H-triple if*

$$\mathcal{I}, \mathcal{V}, [b_1, e_1] \times [b_2, e_2] \models_{ITL^n} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I$$

i.e. for every subrectangle of $[b_1, e_1] \times [b_2, e_2]$ holds H_F^I.

Lemma 1 *Given an arbitrary H-triple $(\mathcal{I}, \mathcal{V}, [b_1, e_1] \times [b_2, e_2])$ such that $b_1 < e_1$ and $b_2 < e_2$ then for every $\pi \in \mathcal{S}$ there is a finite partition in sub-rectangles $[b_1^1, e_1^1] \times [b_2^1, e_2^1], \ldots [b_1^n, e_1^n] \times [b_2^n, e_2^n]$ such that for every rectangle $[b_1^i, e_1^i] \times [b_2^i, e_2^i]$ holds either*

$$\mathcal{I}, \mathcal{V}, [b_1^i, e_1^i] \times [b_2^i, e_2^i] \models_{ITL^n} \lceil v_{[\pi]} \rceil \text{ or}$$
$$\mathcal{I}, \mathcal{V}, [b_1^i, e_1^i] \times [b_2^i, e_2^i] \models_{ITL^n} \lceil v_{[\neg\pi]} \rceil$$

Proof. Let $(x, y) \in [b_1, e_1] \times [b_2, e_2]$. Then by $\mathcal{H}_6$ there exists $x_1 \leq x \leq x_2$ and $y_1 \leq y \leq y_2$ such that

$$\mathcal{I}, \mathcal{V}, [x_1, x] \times [y_1, y] \models_{\mathsf{ITL}^n} \lceil v_{[\pi]} \rceil \vee \lceil v_{[\neg\pi]} \rceil \quad \text{and}$$
$$\mathcal{I}, \mathcal{V}, [x_1, x] \times [y, y_2] \models_{\mathsf{ITL}^n} \lceil v_{[\pi]} \rceil \vee \lceil v_{[\neg\pi]} \rceil \quad \text{and}$$
$$\mathcal{I}, \mathcal{V}, [x, x_2] \times [y_1, y] \models_{\mathsf{ITL}^n} \lceil v_{[\pi]} \rceil \vee \lceil v_{[\neg\pi]} \rceil \quad \text{and}$$
$$\mathcal{I}, \mathcal{V}, [x, x_2] \times [y, y_2] \models_{\mathsf{ITL}^n} \lceil v_{[\pi]} \rceil \vee \lceil v_{[\neg\pi]} \rceil$$

Now $]x_1, x_2[\times]y_1, y_2[$ is an open interval covering (x, y) and the closed interval $[x_1, x_2] \times [y_1, y_2]$ has the desired property. Then by Heine-Borels Theorem there is a finite subset covering $[b_1, e_1] \times [b_2, e_2]$ of this infinite covering. The cases where (x, y) is on the border are handled similarly. This yields the finite partition as required. □

Using this result, for every H-triple $(\mathcal{I}_{\mathsf{ITL}^n}, \mathcal{V}, [b, e])$, we can construct an SC-interpretation $\mathcal{I}_{\mathsf{SC}}$ by defining for every observable X $\mathcal{I}_{\mathsf{SC}}(X)$ to be

$$\mathcal{I}_{\mathsf{SC}}(X)((x,y)) \stackrel{df}{=} \begin{cases} 1 & \text{if there are } x_1, x_2, y_1, y_2 \\ & x_1 \leq x < x_2, y_1 \leq y < y_2 \\ & \text{such that} \\ & \mathcal{I}_{\mathsf{ITL}^n}, \mathcal{V}, [x_1, x_2] \times [y_1, y_2] \\ & \qquad \models_{\mathsf{ITL}^n} \lceil v_{[X]} \rceil \\ 0 & \text{otherwise} \end{cases}$$

This interpretation has the required finite variability property, that each interval can be partitioned into finitely many subintervals such that the interpretation is constant on each subinterval. One can show by induction on the structure of state assertions that for this interpretation $\mathcal{I}_{\mathsf{SC}}$ and every state assertion π holds

$$\mathcal{I}_{\mathsf{SC}}[\![\textstyle\int \pi]\!][c, d] = \mathcal{I}_{\mathsf{ITL}^n}[\![v_{[\pi]}]\!][c, d]$$

Using this result, we can construct for every ITL^n interpretation $\mathcal{I}_{\mathsf{ITL}^n}$ which violates $\Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$ an SC interpretation $\mathcal{I}_{\mathsf{SC}}$ violating F. This proves the following lemma.

Lemma 2 $\models_{SC} F$ *implies* $\models_{ITL^n} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$.

To show the converse implication, let $\mathcal{I}_{\mathsf{SC}}$ be an SC interpretation violating F. Define the violating ITL^n interpretation $\mathcal{I}_{\mathsf{ITL}^n}$ by

$$\mathcal{I}_{\mathsf{ITL}^n}(v_{[\pi]})([b_1, e_1] \times [b_2, e_2]) \stackrel{df}{=} \mathcal{I}_{\mathsf{SC}}[\![\textstyle\int \pi]\!]([b_1, e_1] \times [b_2, e_2]).$$

Using this interpretation and the soundness of axiomatisation, we obtain

Lemma 3 $\models_{ITL^n} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$ *implies* $\models_{SC} F$.

To prove the relative completeness, suppose $\models_{\mathsf{SC}} F$. Then by lemma 2 $\models_{\mathsf{ITL}^n} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$. Take the ITL^n derivation of $\models_{\mathsf{ITL}^n} \Box_{\vec{e}_1} \Box_{\vec{e}_2} H_F^I \Rightarrow F^I$ and replace every occurrence of $v_{[\pi]}$ by $\int \pi$ to obtain an SC derivation. As H_F is a conjunction of instances of SC axioms, it can be easily deduced in SC and therefore using modus ponens we obtain a derivation of F. □

4. Decidable Subsets

Decidable subsets play an import role as they make the implementation of model-checkers possible. For the discrete one-dimensional Duration Calculus there exists the model-checker DCVALID [11]. In this section we present two different types of decidable subsets. One is obtained by imposing restrictions on the class of models and the other by imposing restrictions on the class of formulae. As Duration Calculus is already undecidable in the continuous case, henceforth we assume the time-space-domain to be discrete.

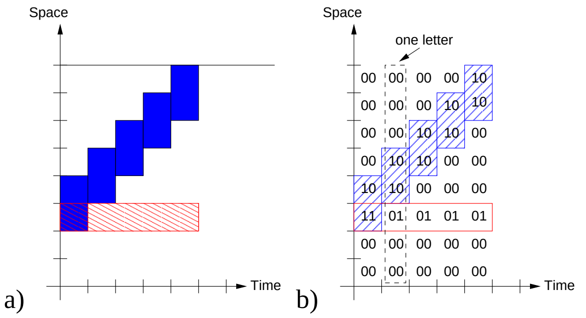

Figure 4. a) Two objects in finite space b) Their representation using a finite alphabet.

4.1. Finite space, infinite time

This first subset $\mathsf{SC}_{\mathsf{fin}}$ imposes a restriction on the class of allowed models. We allow one infinite temporal dimension and require the other spatial dimensions to be finite. For simplicity we illustrate this for only one finite spatial and one infinite temporal dimension. The set of formulae is given by the following EBNF:

$$F ::= \lceil P \rceil \mid F \wedge G \mid \neg F \mid F \langle \vec{e}_x \rangle G \mid F \langle \vec{e}_t \rangle G$$

The approach is sketched in figure 4. As there are only finitely many observables, a configuration for a point in space-time can be represented by a bitvector. All spatial dimensions are finite, so the spatial configuration for a moment in time can be represented by a matrix of bitvectors. The size of the matrix is fixed by the size of space and so we use the set of these matrices as the finite alphabet. The complete decision procedure is given in [13] which can even handle projection onto the temporal oder spatial axes. For brevity we only sketch the construction here. We proceed by inductively constructing a regular language L for a given formula F such that the words in L represent satisfying interpretations for F. Conjunction is constructed by intersection, negation by complementation, temporal chop by concatenation and spatial chop using inverse homomorphisms. So we obtain the following proposition.

Theorem 3 *SC_{fin} is decidable.*

Proof. Let $\{X_1, \ldots, X_p\}$ be the set of observables occuring in F. For discrete space of cardinality m, the set $\mathcal{G}^m \overset{df}{=} \{0,1\}^{m \times p}$ of all $m \times p$ matrices is the set of all possible spatial configurations for one moment in time, i.e. for a matrix $(g_{i,j}) \in \mathcal{G}^m$ $g_{i,j} = 1$ iff the observable X_j is true at point i. We define a family of functions $h^m_{i,j} : \mathcal{G}^m \rightarrow \mathcal{G}^{i-j+1}$ which return from a matrix the submatrix from row i up to row j. For $i > j$ it will return the $0 \times p$ matrix. For a formula F and a spatial cardinality of m, we inductively construct a regular language $\mathcal{L}^m(F)$ over the alphabet $\mathcal{G}^m$ representing all satisfying interpretations for F. Clearly all functions $h^m_{i,j}$ are language homomorphisms. At first we define what it means that one row of a matrix satisfies a state assertion.

$$\begin{aligned}
h^m_{k,k}((g_{i,j})) \models X_q &\overset{df}{\iff} g_{k,q} = 1 \\
h^m_{k,k}((g_{i,j})) \models \neg\pi &\overset{df}{\iff} h^m_{k,k}((g_{i,j})) \not\models \pi \\
h^m_{k,k}((g_{i,j})) \models \pi_1 \wedge \pi_2 &\overset{df}{\iff} h^m_{k,k}((g_{i,j})) \models \pi_1 \\
&\qquad \text{and } h^m_{k,k}((g_{i,j})) \models \pi_2
\end{aligned}$$

Using this terminology, the everywhere expression $\lceil \pi \rceil$ is satisfied by all non-empty sequences of matrices such that in every matrix in every row π holds.

$$\mathcal{L}^m(\lceil \pi \rceil) \overset{df}{=} \begin{cases} \varnothing & \text{if } m = 0 \\ \{(g_{i,j}) \in \mathcal{G}^m \mid & \\ \forall 1 \leq k \leq m : h^m_{k,k}((g_{i,j})) \models \pi\}^+ & \text{otherwise} \end{cases}$$

The constructions for conjunction, negation and temporal chop are straightforward.

$$\begin{aligned}
\mathcal{L}^m(F_1 \wedge F_2) &\overset{df}{=} \mathcal{L}^m(F_1) \cap \mathcal{L}^m(F_2) \\
\mathcal{L}^m(\neg F_1) &\overset{df}{=} \overline{\mathcal{L}^m(F_1)} \\
\mathcal{L}^m(F_1 \langle \vec{e}_t \rangle F_2) &\overset{df}{=} \mathcal{L}^m(F_1) \circ \mathcal{L}^m(F_2)
\end{aligned}$$

A sequence of $m \times p$ matrix satisfies a formula $F_1 \langle \vec{e}_x \rangle F_2$ iff there is an r between 0 and m such that the sequence of the lower r rows satisfies F_1 and the sequence of the upper $m - r$ rows satisfies F_2. To this end, we construct the language of all $r \times p$ sequences that satisfy F_1 use $(h^m_{1,r})^{-1}$ to create all possible extensions to $m \times p$ matrices. This is done for F_2 respectively. The intersection of these languages has the desired property.

$$\begin{aligned}
\mathcal{L}^m(F_1 \langle \vec{e}_x \rangle F_2) \overset{df}{=} \bigcup_{r \in \{0,\ldots,m\}} &\big((h^m_{1,r})^{-1}(\mathcal{L}^r(F_1)) \\
&\cap (h^m_{r+1,m})^{-1}(\mathcal{L}^{m-r}(F_2))\big)
\end{aligned}$$

This definition yields

Lemma 4 $\mathcal{L}(F) = \varnothing \iff$ *F is not satisfiable.*

Obviously $\mathcal{L}(F)$ is a regular language and therefore $\mathsf{SC}_{\mathsf{fin}}$ is decidable. □

Expressivity Although this subset seems to be rather limited, there are several expressions of the original language which can be obtained as abbreviations using the restricted set and the fact that the temporal and spatial domain are discrete.

The terms $\ell_{\vec{e}_i}$ are reobtained as it is impossible in a discrete domain to chop an interval of length 1 into two parts of positive length.

$$\begin{aligned}
\lceil\rceil &\overset{df}{\iff} \neg\lceil 1\rceil\\
\ell_{\vec{e}_i} = 1 \wedge \neg\lceil\rceil &\overset{df}{\iff} \lceil 1\rceil \wedge \neg(\lceil 1\rceil \,\langle\vec{e}_i\rangle\, \lceil 1\rceil)\\
\ell_{\vec{e}_i} = k+1 &\overset{df}{\iff} (\ell_{\vec{e}_i} = k)\,\langle\vec{e}_i\rangle\,(\ell_{\vec{e}_i} = 1)\\
\ell_{\vec{e}_i} > k &\overset{df}{\iff} (\ell_{\vec{e}_i} = k)\,\langle\vec{e}_i\rangle\,\lceil 1\rceil;
\end{aligned}$$

where $k \in \mathbb{N}^+$. The definition of the other operators $\leq$,$\geq$,$<$ is straightforward. As interpretations may only change their value at discrete points the measure $\int P$ can be expressed as follows:

$$\begin{aligned}
\textstyle\int P = 0 &\overset{df}{\iff} \lceil\neg P\rceil \vee \lceil\rceil\\
\textstyle\int P = 1 &\overset{df}{\iff} \textstyle\int P = 0\,\langle\vec{e}_x\rangle\\
&\quad (\textstyle\int P = 0\,\langle\vec{e}_t\rangle\\
&\quad\quad (\lceil P\rceil \wedge \ell_x = 1 \wedge \ell_t = 1)\,\langle\vec{e}_t\rangle\\
&\quad\quad \textstyle\int P = 0)\,\langle\vec{e}_x\rangle\\
&\quad \textstyle\int P = 0\\
\textstyle\int P = k &\overset{df}{\iff} \bigvee_{\substack{k_1,k_2>0\\k_1+k_2=k}} ((\textstyle\int P = k_1)\,\langle\vec{e}_x\rangle\,(\textstyle\int P = k_2)) \vee\\
&\quad \bigvee_{\substack{k_1,k_2>0\\k_1+k_2=k}} ((\textstyle\int P = k_1)\,\langle\vec{e}_t\rangle\,(\textstyle\int P = k_2))
\end{aligned}$$

4.2. Non-alternating chop

Another possibility of deriving a decidable subset is to use the fibrings and dovetailing ideas presented by Gabbay et al [4, 3]. This technique is used to combine two modal logics. To create a structure for the combined logic one uses a structure for the first one and associate to each world a structure for the second logic and so on. The idea is depicted in figure 5 (a). Using this approach a lot of nice properties like axiomatisability and decidability are inherited by the combination.

We need to rule out models like these sketched in 5 (b) where w and w' do not coincide as our main goal is to reason about objects in $\mathbb{N}^n$. To this end, we do not allow chop-alternation. On the outermost nesting level we only allow formulae using $\langle\vec{e}_1\rangle$ nested by formulae using $\langle\vec{e}_2\rangle$ and so on. Additionally, we restrict the interaction of formulae by adding a constraint on the length.

The idea is sketched for the two dimensional case but can be extended to more dimensions. The language of this

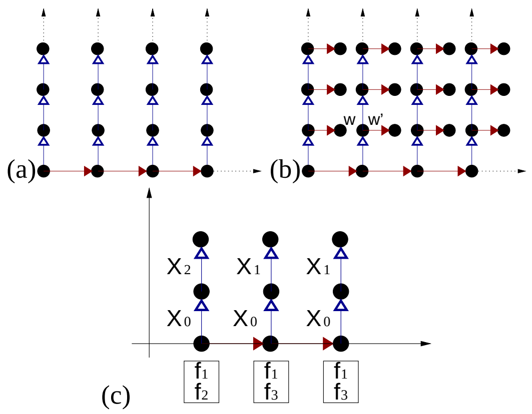

Figure 5. (a) Dovetailing linear modal structures (b) World w and w' may be different. (c) Dovetailing SC

subset $\mathsf{SC}_{\mathsf{nAlt}}$ is the set of formulae F^1 generated by the following EBNF:

$$\begin{aligned}
F^1 &::= F^1_1\,\langle\vec{e_1}\rangle\,F^1_2 \mid F^1_1 \wedge F^1_2 \mid \neg F^1_1 \mid F^2 \wedge \ell_{\vec{e}_1} = 1\\
F^2 &::= \lceil P\rceil \mid F^2_1\,\langle\vec{e_2}\rangle\,F^2_2 \mid F^2_1 \wedge F^2_2 \mid \neg F^2_1
\end{aligned}$$

Let $\delta(F)$ denote the maximal i such that F can be generated from F^i. Although the restriction $F^2 \wedge \ell_{\vec{e}_1} = 1$ appears to be severe, this construction can be used to describe intervals of constant length by using chop. Note, that without this restriction, it is already possible to encode the tiling problem and the resulting subset is undecidable.

The decision procedure constructs inductively regular languages representing fulfilling interpretations.

$\delta(F) = 2$: In this case F is a pure DC formula and we construct the language in the same way as for discrete DC. Let $X_0,\ldots,X_z$ be the boolean observables occurring in F. Then $(x_0,\ldots,x_z) \in \{0,1\}^z$ represents a valuation of these observables for an interval of unit length. Define $\mathcal{L}^2(F)$ inductively by

$$\begin{aligned}
\mathcal{L}^2(\lceil\pi\rceil) &\overset{df}{=} \{(x_0,\ldots,x_z) \mid (x_0,\ldots,x_z) \models \pi\}^+,\\
\mathcal{L}^2(F \wedge G) &\overset{df}{=} \mathcal{L}^2(F) \cap \mathcal{L}^2(G),\\
\mathcal{L}^2(F\,\langle\vec{e}_2\rangle\,G) &\overset{df}{=} \mathcal{L}^2(F) \circ \mathcal{L}^2(G),\\
\mathcal{L}^2(\neg F) &\overset{df}{=} \overline{\mathcal{L}^2(F)}.
\end{aligned}$$

$\delta(F) = 1$: In this case the subformulae of type 2 play the role of the observables. Let $(F_1,\ldots,F_y)$ be the family of subformulae of F with $\delta(F_i) = 2$. Then the vector $(f_1,\ldots,f_y) \in \{0,1\}^y$ can describe which *formulae* are required to hold for an interval of length one.

At first we construct a regular language $\mathcal{L}'$ in the same way as in the above case.

$$\mathcal{L}'(F_i^2 \wedge \ell_{\vec{e}_2} = 1) \stackrel{df}{=} \{(f_1, \dots, f_y) | f_i = 1\}$$
$$\mathcal{L}'(F_1^1 \langle \vec{e}_1 \rangle F_2^1) \stackrel{df}{=} \mathcal{L}(F_1^1) \circ \mathcal{L}(F_2^1)$$
$$\mathcal{L}'(F_1^1 \wedge F_2^1) \stackrel{df}{=} \mathcal{L}(F_1^1) \cap \mathcal{L}(F_2^1)$$
$$\mathcal{L}'(\neg F^1) \stackrel{df}{=} \overline{\mathcal{L}(F^1)}$$

Different from the simple case, the language $\mathcal{L}'$ does not represent the set of satisfying interpretations. For example the requirement that F_1 and F_2 hold for the same interval may not be satisfiable. So we have to ensure that

- for each vector $(f_1, \dots, f_y)$ there is an interpretation such that exactly those formulae F_i hold where $f_i = 1$ and
- there is a length k such that for all vectors there is a satisfying interpretation of this length.

Let $\Sigma = \{(f_1, \dots, f_y) \mid f_i \in \{\text{true}, \text{false}\} \text{ for all } 1 \leq i \leq y\}$ the set of all vectors. For one vector $(f_1, \dots, f_y)$ we define its associated language by

$$\mathcal{L}^2((f_1, \dots, f_y)) \stackrel{df}{=} \bigcap_{i \in \{1, \dots y\} | f_i = true} \mathcal{L}^2(F_i) \cap \bigcap_{i \in \{1, \dots y\} | f_i = false} \overline{\mathcal{L}^2(F_i)}.$$

This languages represents all interpretations that make exactly those formulae true which are indicated by the vector.

Definition 3 *Let $\sharp$ be an arbitrary symbol and $h_\sharp : \Sigma \to \{\sharp\}$ be the homomorphism that simply replaces every letter by $\sharp$.*

Definition 4 *A subset $\Pi \subseteq \Sigma$ is called* consistent *iff*

$$\bigcap_{(f_1, \dots, f_y) \in \Pi} h_\sharp(\mathcal{L}^2((f_1, \dots, f_y))) \neq \varnothing$$

This definition ensures the two requirements stated above. So we can define

$$\mathcal{L}(F) \stackrel{df}{=} \mathcal{L}'(F) \cap (\bigcup_{\substack{\Pi \subseteq \Sigma \\ \Pi \text{ is consistent}}} \Pi^*).$$

Using this definition one obtains

Lemma 5 *$\mathcal{L}(F) \neq \varnothing$ iff F is satisfiable*

and as all these constructions can be done effectively this proves the following theorem.

Theorem 4 *$\mathsf{SC}_{\mathsf{nAlt}}$ is decidable.*

Example 6 *These constructions are illustrated in 5 (c). In order to decide*

$$F \stackrel{df}{=} (F_1 \wedge \ell_{\vec{e}_1} = 1 \langle \vec{e}_1 \rangle F_1 \wedge \ell_{\vec{e}_1} = 1 \langle \vec{e}_1 \rangle F_1 \wedge \ell_{\vec{e}_1} = 1) \wedge (F_2 \wedge \ell_{\vec{e}_1} = 1 \langle \vec{e}_1 \rangle F_3 \wedge \ell_{\vec{e}_1} = 1 \langle \vec{e}_1 \rangle F_3 \wedge \ell_{\vec{e}_1} = 1)$$

with

$$F_1 \stackrel{df}{=} \lceil X_1 \rceil \langle \vec{e}_2 \rangle \text{true},$$
$$F_2 \stackrel{df}{=} \text{true} \langle \vec{e}_2 \rangle \lceil X_2 \rceil,$$
$$F_3 \stackrel{df}{=} \text{true} \langle \vec{e}_2 \rangle \lceil X_3 \rceil.$$

The word $(1,1,0)(1,0,1)(1,0,1)$ is in $\mathcal{L}'(F)$ and as the alphabet is consistent also in $\mathcal{L}(F)$. Therefore the models for F_1, F_2, F_3 can be combined to form a model for F.

Expressivity Like in $\mathsf{SC}_{\mathsf{fin}}$ operators can be reobtained in $\mathsf{SC}_{\mathsf{nAlt}}$. At first we give definitions for formulae of type 2 which are to be used in the context of "$\wedge\, \ell_{\vec{e}_1} = 1$". We use the superscript ^{2} here to stress this restriction.

$$\text{true}^2 \stackrel{df}{\iff} \lceil 1 \rceil^2 \vee \neg \lceil 1 \rceil^2$$
$$\ell_{\vec{e}_2}^2 = 0 \stackrel{df}{\iff} \neg \lceil 1 \rceil^2$$
$$\ell_{\vec{e}_2}^2 = 1 \stackrel{df}{\iff} \lceil 1 \rceil^2 \wedge \neg(\lceil 1 \rceil^2 \langle \vec{e}_2 \rangle \lceil 1 \rceil^2)$$
$$\ell_{\vec{e}_2}^2 = k+1 \stackrel{df}{\iff} (\ell_{\vec{e}_2}^2 = k) \langle \vec{e}_2 \rangle (\ell_{\vec{e}_2}^2 = 1)$$
$$\ell_{\vec{e}_2}^2 > k \stackrel{df}{\iff} (\ell_2^2 = k) \langle \vec{e}_2 \rangle \lceil 1 \rceil^2$$
$$\int^2 P = 0 \stackrel{df}{\iff} \lceil \neg P \rceil^2 \vee \ell_{\vec{e}_2}^2 = 0$$
$$\int^2 P = 1 \stackrel{df}{\iff} \int^2 P = 0 \langle \vec{e}_2 \rangle \lceil P \rceil^2 \wedge \ell_{\vec{e}_2}^2 = 1 \langle \vec{e}_2 \rangle \int^2 P = 0$$
$$\int^2 P = k+1 \stackrel{df}{\iff} \int^2 P = k \langle \vec{e}_2 \rangle \int^2 P = 1$$

For formulae of type 1 the definitions are more complicated. At first true can be defined in the standard way.

$$\text{true} \stackrel{df}{\iff} (\lceil 1 \rceil \wedge \ell_{\vec{e}_1} = 1) \vee \neg(\lceil 1 \rceil \wedge \ell_{\vec{e}_1} = 1)$$

As $\ell_{\vec{e}_1}$ is nearly a primitive in $\mathsf{SC}_{\mathsf{nAlt}}$, it can be defined as follows:

$$\ell_{\vec{e}_1} = 1 \stackrel{df}{\iff} (\text{true}^2) \wedge \ell_{\vec{e}_1} = 1$$
$$\ell_{\vec{e}_1} = k+1 \stackrel{df}{\iff} (\ell_{\vec{e}_1} = k) \langle \vec{e}_1 \rangle (\ell_{\vec{e}_1} = 1)$$

The measure $\int P$ is zero iff there is no subinterval of length 1 on which the measure is not zero. Therefore this can be defined using the type 2 formula $\int^2 P = 0$.

$$\int P = 0 \stackrel{df}{\iff} \neg\big(\text{true} \langle \vec{e}_1 \rangle ((\neg(\int^2 P = 0)) \wedge \ell_{\vec{e}_1} = 1) \langle \vec{e}_1 \rangle \text{true}\big)$$

Using the same idea, we can define $\int P = 1$.

$$\int P = 1 \overset{df}{\iff} \int P = 0 \langle \vec{e}_1 \rangle \\ (\int^2 P = 1 \wedge \ell_{\vec{e}_1} = 1) \langle \vec{e}_1 \rangle \int P = 0$$

On an interval of length m the measure $\int P$ equals k iff it is equal to k_1 on the leftmost subinterval of length $m-1$, is equal to k_2 on the rightmost subinterval of length 1 and $k = k_1 + k_2$.

$$\int P = k \overset{df}{\iff} \bigvee_{\substack{k_1,k_2 \in \mathbb{N}_0 \\ k_1+k_2=k}} \Big(\int P = k_1 \langle \vec{e}_1 \rangle \\ (\int^2 P = k_2 \wedge \ell_{\vec{e}_1} = 1) \Big)$$

5 Conclusion

In this paper we investigated properties of a multi-dimensional extension of duration calculus. We show that it is not axiomatisable and therefore not decidable. Nevertheless we can give an axiomatisation relative to a n-dimensional interval temporal logic. Tool-support is crucial when thinking of applications of such a formalism in practice. So decidable subsets play an important role. We present two different ones. One is retrieved by restricting the models the other by restricting the formulae.

Related work There is a lot of work done in the area of spatio-temporal logics, for example formalisms based on modal logic like in [2] or spatio-temporal logics [5] based on the Region Connection Calculus by Randell, Cui and Cohn [12] with a lot of applications in AI. But these approaches do not allow to measure time and space. A quantitative measure is possible in [14] but it does not consider time and space explicitly. It uses general metric spaces as models. Other approaches like [9] adopt the π-calculus [10] or ambient-calculus notion of mobility for a different application domain.

Perspectives We would like to apply this formalism to several case studies to derive a set of lightweight rules that make the handling of this formalism in practice easier. To give tool support, this should be accompanied by extending and implementing the decision procedures found so far.

Acknowledgements The author thanks E.-R. Olderog and the members of the "Correct System Design" group, especially J. Hoenicke and M. Möller for fruitful discussions on this topic.

References

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] B. Bennett and A. Cohn. Multi-Dimensional Multi-Modal Logics as a Framework for Spatio-Temporal Reasoning. *Applied Intelligence*, 17(3):239–251, 2002.

[3] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier, 2003.

[4] D. M. Gabbay. *Fibring Logics*. Oxford University Press, Oxford, 1999.

[5] A. Galton. Towards a qualitative theory of movement. In *Spatial Information Theory*, pages 377–396, 1995.

[6] D. Giammarresi and A. Restivo. *Handbook of Formal Languages – Beyond Words*, volume 3, chapter Two-Dimensional Languages, pages 215–267. Springer, 1997.

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

[8] M. R. Hansen and Zhou Chaochen. *Duration Calculus: A Formal Approach to Real-Time Systems*. EATCS: Monographs in Theoretical Computer Science. Springer, 2004.

[9] S. Merz, M. Wirsing, and J. Zappe. A Spatio-Temporal Logic for the Specification and Refinement of Mobile Systems. In M. Pezzè, editor, *FASE 2003, Warsaw, Poland*, volume 2621 of *LNCS*, pages 87–1014. Springer, 2003.

[10] R. Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, 1999.

[11] P. Pandya. Specifying and deciding qauntified discrete-time duration calculus formulae using dcvalid. Technical report, Tata Institute of Fundamental Research, 2000.

[12] D. A. Randell, Z. Cui, and A. Cohn. A Spatial Logic Based on Regions and Connection. In B. Nebel, C. Rich, and W. Swartout, editors, *KR'92.*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.

[13] A. Schäfer. A Calculus for Shapes in Time and Space. In Z. Liu and K. Araki, editors, *ICTAC 2004*, volume 3407 of *LNCS*, pages 463–478. Springer, 2005.

[14] F. Wolter and M. Zakharyaschev. Reasoning about distances. In G. Gottlob and T. Walsh, editors, *IJCAI-03, Acapulco, Mexico, August 9-15, 2003*, pages 1275–1282. Morgan Kaufmann, 2003.

[15] Zhou Chaochen, C. Hoare, and A. Ravn. A calculus of durations. *IPL*, 40(5):269–276, 1991.