Marking time

Kamal Lodaya

The Institute of Mathematical Sciences, C.I.T. Campus, Chennai 600 113, India.
Email: kamal@imsc.res.in

Tense logics. The traditional viewpoint of logics of time is as a kind of modal logic, with Kripke models where the accessibility relation is deemed to specify properties particular to time. We call this tense logic after a classic survey by John Burgess in the *Handbook of philosophical logic* [1984]. We begin by changing the tense logic models to incorporate duration.

Next, we discuss the point-based and interval-based versions of time. These were studied in Johan van Benthem's book *The logic of time* and in several papers appearing around the same time: Vilain [1982], and Allen [1983], developed an interval algebra; Moszkowski and Manna [1983], as well as Schwartz, Melliar-Smith and Vogt [1983], advocated interval-based reasoning for hardware. The prohibitive complexity of logics which evaluate propositions at intervals was established by Halpern and Shoham [1991]. Many fragments have been shown to be undecidable (cf. Lodaya [2000]). Goranko, Montanari, Sciavicco and Bresolin recently showed [2004, 2007] that a few are decidable by exploiting their resemblance to the point-based logic. Interval logic has made a resurgence, even making it into industry standards like PSL/Sugar: Vardi [2006] gives a picture of the history.

We expand the discussion by generalizing from durations to arbitrary measurements. Two logics, one point-based and another interval-based, are presented. We sketch a completeness proof for the point-based logic. We also have a brief section on expressive completeness of these logics.

Temporal and dynamic logics. Zohar Manna and Amir Pnueli [1992] viewed linear time models as *runs* generated from a finite transition system since they were interested in efficient algorithms for verifying time properties. The Kripke frames were fixed to be the natural numbers, or an initial segment. We will use the name temporal logic for this "informatic" approach to time, as Manna and Pnueli did. This transition from tense logic to temporal logic, with automata theory playing a constructive rôle, is detailed in the chapter by Ian Hodkinson in the second volume of the book *Temporal logic* [2000].

Mihir K. **Chakraborty**, Benedikt **Löwe**,
Madhabendra Nath **Mitra**, Sundar **Sarukkai** (*eds.*)
Logic, Navya-Nyaya & Applications.
An Homage to Bimal Krishna Matilal.
College Publications London 200? [Studies in Logic ?]; p. 1-19

A duration calculus was developed early on by Zhou Chaochen, Tony Hoare and Anders Ravn [1991] to reason about timed systems. The book by Zhou and Michael Hansen [2004] is a good reference. Rajeev Alur, David Dill and Tom Henzinger [1994, 1993] developed the "informatic" approach to duration by extending the Manna-Pnueli temporal logic to timed systems.

Again we generalize from durations to arbitrary measurements, and instead of a temporal logic we present a dynamic logic. Decidability is proved for a future fragment, not for arbitrary measurements but only for durations, and only when the models are restricted to be finite.

This article. The purpose of this article is expository: to use these informatic ideas and develop tense and temporal logic afresh, this time with measurement. Many of the definitions and results are new (for instance, we have never seen a dynamic logic with measurement modalities before), but they are small generalizations of what has appeared in the literature.

It is not our aim to survey the field of logics dealing with time. Burgess [1984], and more recently Goranko, Montanari and Sciavicco [2004] and Vardi [2006] provide many references. The two volumes of *Temporal logic* edited by Gabbay, Hodkinson, Reynolds and Finger offer a reasonably up-to-date compendium of technical details. The articles by Galton [1995], Hayes [1995] and van Benthem [1995] have a discussion of linguistic details.

Acknowledgements. I thank the organizers of the *Logic, Navya-Nyāya and Applications* conference in honour of Bimal Krishna Matilal, January 2007, for inviting me to speak on the occasion, and the hospitality of Jadavpur University where the conference was held. An earlier version of the talk was presented at the workshop on *Advances and Issues in Timed Systems* during *FSTTCS* XXVI, Kolkata, December 2006. I thank Supratik Chakraborty and Deepak D'Souza for inviting me to speak there, and the Indian Statistical Institute for their hospitality.

Paritosh Pandya and I collaborated itinerantly on a paper on measurement logics, which finally got written [2006]. Many of the definitions in the present article arise from that work. As a consequence of his expertise on timed systems, Paritosh had to put up with a dense barrage of questions during the interval I was preparing for the Kolkata talks, towards the end of 2006. At the Matilal conference a little later, Johan van Benthem expressed interest in the development of sampled time models. R. Ramanujam started off a lively discussion on "time" in timed systems while I was writing this article. Sunil Simon enlightened me on the fine points of alternating timed automata. To all of them, thank you.

1 Duration frames and measurement models

Definition 1 (Dutertre). A duration domain $(D, +, 0, <)$ is a linearly ordered monoid which is cancellative (if $x + y = x + z$ or $y + x = z + x$ then $y = z$) and zerosumfree (if $x + y = 0$ then $x = y = 0$).

In this article we will only work with duration domains which are Abelian. Common examples of duration domains are the natural numbers and the nonnegative reals.

Definition 2. A (point) duration frame $\mathcal{T} = (T, \prec, d)$ is a nonempty linear order $\mathcal{T} = (T, \prec)$ (the underlying flow of time) with a symmetric order-preserving distance function d from $T \times T$ into a duration domain D.

We will also define interval duration frames. The general definition follows the one in van Benthem's book [1983].

Definition 3. An interval duration frame $\mathcal{I} = (I, \subset, \ll, d)$ is a nonempty poset $(I, \subset)$ with greatest lower bounds, together with a partial order $\ll$ and a distance function $d : I \to D$ into a duration domain D which are monotonic with respect to $\subset$. That is, if $w \subset x \ll y \supset z$ then $w \ll y$, $d(w) \leq d(x)$ and $d(y) \geq d(z)$.

Given a point frame $\mathcal{T} = (T, \prec, d)$, we can construct an interval frame $Int(\mathcal{T}) = (I, \subset, \ll, d)$ by letting I be the nonempty convex subsets of T, $\subset$ be inclusion, $x_1 \ll x_2$ iff for every t_1 in x_1, t_2 in x_2, $t_1 \prec t_2$ and $d(x) = d(b, e)$ where b and e are the beginning and ending points (or limit points) of the interval x. This interval frame is also atomic: for every x there is $x_1 \subset x$, such that if $x_0 \subset x_1$ then $x_0 = x_1$.

Given an atomic interval frame $\mathcal{I} = (I, \subset, \ll, d)$, we can construct a point frame $Pt(\mathcal{I}) = (T, \prec, d)$ by letting T be the set of atoms of I, $\prec$ be $\ll$ restricted to T and $d(b, e) = \inf\{d(x) \mid b, e \subset x\}$.

The theorem below was proved for tense frames, but extending it to duration frames is not difficult.

Theorem 4 (van Benthem). An atomic interval frame $\mathcal{I}$ is isomorphic to $Int(Pt(\mathcal{I}))$.

Let us consider a more specific example. If we take $\mathcal{T}$ to be a strict linear order, we can define its intervals to be the usual "open intervals" $(t_1, t_2) = \{t \mid t_1 \prec t \prec t_2\}$. It is an easy exercise to list the 5 possible point-interval relations (Vilain [1982]). Hamblin [1969] and Allen [1983] showed that all 13 interval-interval relations are definable using $\subset$ and $\ll$ (the first two below). The converses of these six and the identity relation make up the total.

- (t, u) *during* (v, w) if $v \prec t$ and $u \prec w$,
- (t, u) *before* (v, w) if $u \prec v$.
- (t, u) *overlaps* (v, w) if $t \prec v \prec u \prec w$.
- (t, u) *meets* (v, w) if $u = v$.
- (t, u) *begins* (v, w) if $t = v$ and $u \prec w$.
- (t, u) *ends* (v, w) if $v \prec t$ and $u = w$.

1.1 A signature of measurements

Let $Prop$ be a set of propositions. Let $\Sigma = \{m_1, m_2, \dots\}$ be a signature of measurement function symbols (of arity 2). Σ always contains the distinguished function ℓ which will be used to measure the length of an interval of time. We will abbreviate the signature $\{\ell\}$ to ℓ.

Definition 5. A measurement model $M = (\mathcal{T}, \theta)$ is a duration frame with a behaviour $\theta : (Prop \to T \to \{0, 1\}) \times (\Sigma \to (T \times T) \to D)$ such that $\theta(\ell)[b, e] = d(b, e)$. An interval measurement model is given by $M = (Int(\mathcal{T}), \theta)$.

A behaviour consists of a valuation, a boolean function of time which we write as $\theta(p)$, together with an interpretation of the measurement signature, with $\theta(m)[b, e]$ giving the value of the measurement function $m \in \Sigma$ on the interval $[b, e]$. The behaviour is defined in this way so that it allows Σ to depend on the propositions: the *Duration calculus* book [2004] has examples of measurements $\int p$, which give the total duration for which a proposition p holds in a given interval.

In this article we will only work with measurement functions which are symmetric. We can impose further conditions on the measurement functions, such as making them additive, order-preserving or anti-order-preserving, as required. Moreover, we require that the measurement ℓ is always interpreted by the distance function.

2 Measurement logics

The formulae of point measurement logic, defined below, allow tense modalities and future and past modalities which are simple generalizations of the usual tense modalities to measurement. This logic is a generalization of the metric tense logic defined by Burgess [1984] and *MTL* defined by Koymans [1990], which only dealt with the length signature ℓ.

Throughout this article we will use χ as a parameter for a set of comparison operations, for example $Punct = \{<, =, >, \leq, \geq\}$ is called the set of punctual comparisons. $Eq = \{=\}$ is the set of equality comparisons. The set of *Weak* comparisons is defined so that equality comparisons are not definable in the logic, for example $\{\leq, >\}$. This use of parameters is from a paper with Pandya [2006].

In the syntax below, the metavariable m gives the value of the function m during the interval of interest. The actual value of m is not accessible in the syntax, but only a guard: a comparison of m with a constant c. We use $-m$ to denote that the interval is to be oriented going into the past.

Definition 6 (Point measurement logic $\chi MTL[\Sigma]$).

$$\begin{array}{rcl}\alpha & ::= & p \in Prop \mid \neg\alpha \mid \alpha \vee \beta \mid \alpha\,\mathcal{U}\,\beta \mid \alpha\,\mathcal{S}\,\beta \mid \\ & & \langle -m \sim c\rangle\alpha \mid \langle m \sim c\rangle\alpha,\ m \in \Sigma,\ \ \sim \in \chi,\ c \in D\end{array}$$

Satisfaction is inductively defined as usual.

$M, t \models p$ iff $\theta(p)[t] = 1$
$M, t \models \alpha\,\mathcal{U}\,\beta$ iff $\exists v \prec t : M, v \models \beta$ and $\forall l : t \prec u \prec v : M, v \models \alpha$
$M, t \models \alpha\,\mathcal{S}\,\beta$ iff $\exists r \prec t : M, r \models \beta$ and $\forall s : r \prec s \prec t : M, r \models \alpha$
$M, t \models \langle -m \sim c\rangle\alpha$ iff $\exists s \prec t : \theta(m)[s, t] \sim c$ and $M, s \models \alpha$
$M, t \models \langle m \sim c\rangle\alpha$ iff $\exists u \succ t : \theta(m)[t, u] \sim c$ and $M, u \models \alpha$.

We define the future and past modalities using $\mathcal{U}$ and $\mathcal{S}$ (e.g. $\Diamond\alpha \stackrel{\text{def}}{=} true\,\mathcal{U}\,\alpha$), but they are also definable using the length operators, e.g. $\Diamond\alpha$ is $\langle \ell > 0\rangle\alpha$ or $\langle \ell \geq 0\rangle\alpha$, depending on whether the modality is to be strict or not. We take the comparisons $<, =, >$ as basic and $\leq, \geq$ as abbreviations.

It is also possible to define in the logic until and since operations which specify a measurement comparison. For example, if the requirement β should occur after a measurement $\geq c$ and α is to hold until then, this can be written $[m < c]\alpha \wedge \langle m = c\rangle(\alpha\,\mathcal{U}\,\beta)$. If the measurement is $\leq c$, the formula can be written as $(\alpha\,\mathcal{U}\,\beta) \wedge \langle m \leq c\rangle\beta$.

Here is an example of reasoning, adapted from Burgess [1984].

Suman: Have you heard? Jagan is going to Alabama this September.	$\langle m = 9\rangle jga$
Sameen: He won't get in without a visa. Has he remembered to apply for one?	$\neg\Diamond(jga \wedge \neg\diamondminus jgv)$ $\neg\diamondminus jav \wedge \neg jav$
Suman: Not yet, as far as I know.	$jgv \supset \langle -m \leq 2\rangle jav$
Sameen: Visa queues might even take a month to clear. He'll have to do so by July.	$\therefore \langle m \leq 7\rangle jav.$

Here are some well known axioms for validity of tense logic (see Burgess [1982, 1984]), recast into the measurement framework. We mostly provide the future axioms, leaving the reader to supply the mirror image axioms for the past. There are some specific axioms for the comparisons.

$[m \sim c](\alpha \supset \beta) \supset ([m \sim c]\alpha \supset [m \sim c]\beta),$
$[-m \sim c](\alpha \supset \beta) \supset ([-m \sim c]\alpha \supset [-m \sim c]\beta)$ *K*

$\alpha \supset [\ell \sim c]\langle -\ell \sim c\rangle\alpha, \quad \alpha \supset [-\ell \sim c]\langle \ell \sim c\rangle\alpha$ *symmetry*

$\langle m < c_1\rangle\alpha \supset \langle m < c_1 + c_2\rangle\alpha, \ \langle m > c_1 + c_2\rangle\alpha \supset \langle m > c_1\rangle\alpha$ *monotonicity*

$\Diamond\alpha \equiv \langle m < c\rangle\alpha \vee \langle m = c\rangle\alpha \vee \langle m > c\rangle\alpha$ *linearity*

$\langle \ell = c\rangle\alpha \supset [\ell = c]\alpha$ *unique length*

$\alpha \supset \langle m = 0\rangle\alpha, \quad \langle \ell = 0\rangle\alpha \supset \alpha$ *reflexivity*

$\langle m \sim c_1\rangle\langle m \sim c_2\rangle\alpha \supset \langle m \sim c_1 + c_2\rangle\alpha$ *transitivity*

$\langle m \sim c_1 + c_2\rangle\alpha \supset \langle m \sim c_1\rangle\langle m \sim c_2\rangle\alpha$ *density*

$\langle m \leq c_1\rangle\alpha \wedge \langle m \leq c_1 + c_2\rangle\beta \supset$ *connectedness*
$\langle m \leq c_1\rangle(\alpha \wedge \langle m \leq c_1 + c_2\rangle\beta) \vee \langle m \leq c_1\rangle(\alpha \wedge \beta) \vee \langle m \leq c_1\rangle(\beta \wedge \langle m \leq c_1\rangle\alpha)$

$\alpha\,\mathcal{U}\,\beta \equiv \beta \vee ((\alpha \wedge \alpha\,\mathcal{U}\,\beta)\,\mathcal{U}\,\beta)$ *until*

$\alpha\,\mathcal{U}\,false \supset false$ *unless*

Completeness for the tense fragment is claimed in Burgess [1984, Section 6.1]. We stretch the claim to the measurement setting below. For completeness of the fragment with equality comparisons and duration domains which are ordered abelian groups, see Montanari and de Rijke [1997].

Claim 7. There is a complete axiomatization for $\chi MTL[\Sigma]$.

A formula mentions finitely many measurement functions $\Sigma_0 \subseteq \Sigma$ and finitely many constants $D_0 \subseteq D$ in finitely many guards. We always include ℓ in Σ_0. The guards implicitly divide the product duration domain $D_0^{|\Sigma_0|}$

into finitely many regions $r_1, \ldots, r_n$. For example a region might be $5 < \ell < 7, m_1 > 8, m_2 = 3$. The idea of regions is from Alur and Dill [1994].

Define for each region r_i, an accessibility relation $\prec_i$ which is compatible with the constraints in the region. For example, the accessibility relation for our example region will ensure that for every formula of the kind $[\ell > 5]\alpha$, $[\ell < 7]\alpha$, $[m_1 > 8]\alpha$ or $[m_2 = 3]\alpha$ in Γ, $\alpha \in \Delta$. Using the connectedness axioms, we can show that at least one of the $\prec_i$ relations will hold between any pair of mcs Γ, Δ.

Using these ideas, a Henkin construction can be performed, ensuring that a linear order is maintained. We spell out one detail in the lemma below which provides an mcs satisying a future measurement requirement. It illustrates that modalities like $\langle 5 < \ell < 7 \wedge m_1 > 8 \wedge m_2 = 3\rangle\alpha$, or even those which check that a measurement lies in an interval such as $\langle m \in [b, e]\rangle\alpha$, are not required for proving completeness.

Lemma 8. If a maximal consistent set Γ has $\langle m \sim c\rangle\alpha$ then there is a compatible region r_i and a maximal consistent set $\Delta \succ_i \Gamma$ containing α.

Proof. It is sufficient to show that for some $d \in D_0$, the set of formulae $\Gamma^- = \{\gamma \mid [m \sim c]\gamma \wedge [\ell \sim d]\gamma \wedge \langle \ell \sim d\rangle\alpha \in \Gamma\} \cup \{\alpha\}$ is consistent. The same argument will be repeated for the other measurements in Σ_0, and the resulting consistent set will be expanded to a maximal consisistent set Δ using a Lindenbaum lemma. The comparisons between m and c, ℓ and d and so on constitute the region r_i. Hence $\Delta \succ_i \Gamma$ by construction.

So suppose Γ^- is not consistent. Then it has a finite inconsistent subset whose conjunction we denote by $\widehat{\gamma} \wedge \alpha$. By supposition and the K axiom, $\langle m \sim c\rangle(\widehat{\gamma} \wedge \alpha) \in \Gamma$.

Consider the smallest d_1 in D_0 under the duration order. By the linearity axiom, one of $\langle \ell < d_1\rangle(\widehat{\gamma} \wedge \alpha), \langle \ell = d_1\rangle(\widehat{\gamma} \wedge \alpha), \langle \ell > d_1\rangle(\widehat{\gamma} \wedge \alpha)$ is in Γ.

Suppose, for example, it is not the first two but the third. Then Γ also has $[\ell \leq d_1]\neg(\widehat{\gamma} \wedge \alpha)$.

Now we proceed to the next d_2 in the D_0 ordering. Using monotonicity and linearity, we ask whether $\widehat{\gamma} \wedge \alpha$ lies within the interval (d_1, d_2) or at d_2 or beyond.

Continuing in this way, we will home in on a suitable d in D_0 and arrive at a contradiction with the consistency of Γ. q.e.d.

2.1 Interval measurement logic

An interval measurement logic can also be defined. The logic below was defined in a paper with Pandya [2006] and is used here because it matches a first order logic, as will be seen later. As in the case of the point logic, we use χ as a parameter for a set of comparison operations.

Definition 9 (Guarded interval measurement logic, $\chi GIML[\Sigma]$).

$$\phi ::= \lceil p \rceil,\ p \in Prop \mid \neg\phi \mid \phi \vee \psi \mid \phi;\psi \mid \langle -m \sim c\rangle\psi \mid \langle m \sim c\rangle\psi,\ \ \sim \in \chi,\ c \in D$$

The satisfaction relation is inductively defined below. Note that propositions are evaluated at points and lifted to intervals by making them "hereditary" (van Benthem [1983]).

$$M,[b,e] \models \lceil p \rceil \text{ iff } \forall t : b \prec t \prec e : \theta(p)[t] = 1$$
$$M,[b,e] \models \phi;\psi \text{ iff } \exists z : b \prec z \prec e : M,[b,z] \models \phi \text{ and } M,[z,e] \models \psi$$
$$M,[b,e] \models \langle -m \sim c\rangle\psi \text{ iff } b = e \text{ and } \exists z \prec b : \theta(m)[z,b] \sim c \text{ and } M,[z,b] \models \psi$$
$$M,[b,e] \models \langle m \sim c\rangle\psi \text{ iff } b = e \text{ and } \exists z \succ e : \theta(m)[e,z] \sim c \text{ and } M,[e,z] \models \psi$$

Dutertre [1995] gave an axiomatization of first order interval tense logic. Here are some of his axioms which are applicable in a propositional setting. We do not repeat the axioms for the measurement modalities from the point version. ε stands for the formula $\langle \ell = 0\rangle true$.

$$(\phi \vee \psi);\chi \supset \phi;\chi \vee \psi;\chi, \quad \phi;(\psi \vee \chi) \supset \phi;\psi \vee \phi;\chi \qquad K$$

$$\varepsilon;\phi \equiv \phi \equiv \phi;\varepsilon \qquad \textit{reflexivity}$$

$$(\phi;\psi);\chi \equiv \phi;(\psi;\chi) \qquad \textit{transitivity}$$

Dutertre's completeness proof [1995] is a first order logic Henkin construction. It needs to be examined to pull out a completeness proof for the quantifier-free version of interval logic that we are considering here.

2.2 Decidability

The decidability of measurement logic depends on the comparisons used. Alur and Henzinger [1993] showed that *Punct*-$MTL[\ell]$ validity is undecidable. Their proof uses Eq comparisons of the form $\ell = c$ for $c > 0$. *Weak*-$MTL[\ell]$ was shown decidable by Alur, Feder and Henzinger [1996].

Theorem 10 (Alur, Feder and Henzinger). Validity of *Punct*-$MTL[\Sigma]$ formulae is undecidable, but for *Weak*-$MTL[\ell]$ formulae it is decidable.

The algorithmic situation is no better for interval models.

Theorem 11 (with Pandya). The validity of *Punct-GIML*$[\Sigma]$ formulae is not decidable, but it is decidable for *Weak-GIML*$[\ell]$ formulae.

Proof. Both proofs are by translation. $\chi MTL[\Sigma]$ formulae can be coded into $\chi GIML[\Sigma]$, yielding the undecidability. The decidability follows from that of the first order fragment *Weak-GF*$[\ell]$, defined in the next section, which was proved by Hirshfeld and Rabinovich [2005]. As we will see, *Weak-GIML*$[\ell]$ is expressively complete for this fragment, hence the decidability can be transferred. q.e.d.

3 Expressive completeness

Kamp [1968] introduced a new dimension to tense logic by relating it to the first order theory of linear order with monadic predicates $FO[<]$. Specifically, he showed that tense logic with the binary modalities $\mathcal{U}$ (until) and $\mathcal{S}$ (since) has the same expressive power as three-variable first order logic, which in turn is as expressive as full $FO[<]$. Kamp's work was extended by Stavi to all linear orders. The first volume of the book *Temporal logic* [1994] has a detailed treatment of Kamp's theorem.

Kamp's syntactic techniques were used by Venema [1990] to establish an expressiveness result for interval tense logic with respect to three-variable first order logic. Just as Kamp had to, Venema showed that binary "chop" modalities are needed. If propositions are evaluated at points (as in the previous section), this again yields full $FO[<]$.

To extend these ideas to measurement, observe that the semantics of all the logics we have considered translate into a guarded fragment of first order logics over linear orders extended with measurement functions $FO[<, \Sigma]$.

Definition 12. Let χ be a given set of length comparisons. $\chi GF[\Sigma]$ is the logic which extends $FO[<]$ by the χ-guarded quantifier $\phi(t_0) = \exists t(G(t_0, t) \wedge \psi(t_0, t))$, where ψ is a formula with at most two free variables t_0 and t, and the guard G is a boolean combination of comparisons from the set χ.

In earlier work, Hirshfeld and Rabinovich conjectured that such a fragment does not have an expressively complete modal logic. Recently, Pandya and I refuted this conjecture [2006]. We use all of Venema's chop modalities.

Theorem 13 (with Pandya). An extension of the logic $\chi GIML[\Sigma]$ is expressively complete for the corresponding guarded fragment $\chi GF[\Sigma]$.

Hirshfeld and Rabinovich [2005] defined a smaller fragment by using a point guarded quantifier $\phi(t_0) = \exists t(G(t_0, t) \wedge \psi(t))$, where ψ is a formula with at most *one* free variable t, and the guard G is an *atomic* comparison. An induction shows that the logic $\chi MTL[\Sigma]$ is expressively complete with respect to the corresponding point guarded fragment.

Extension of these results to a monadic second order framework, with a matching extension of the modal logic, for example by propositional quantification, can be added on.

4 Discrete models and sampled time

Manna and Pnueli [1992] shifted attention to the question of *verification* of satisfaction: given a model M, a point t and a formula α, how do we check whether $M, t \models \alpha$? This leads to a somewhat trivial-sounding question, how is the model M to be presented to an algorithm?

Let $Prop$ be a finite set of propositions, Σ a finite signature of additive measurement functions and D a duration domain. We fix an ordering of the function symbols in the signature Σ beginning with the length ℓ. A model M with a discrete linear order as a frame can be represented as a pair of finite words $v = v_0 v_1 \ldots v_n$ and $w = w_0 w_1 \ldots w_n$, or a pair of infinite words $v = v_0 v_1 \ldots$ and $w = w_0 w_1 \ldots$ over the alphabets $D^{|\Sigma|}$ and $\wp(Prop)$ respectively. Since the Kripke frame ($\mathbf{N}$ or an initial prefix $[0..n]$) is fixed by the words, this "word model" consists only of a behaviour.

Definition 14. A sampled time behaviour is a pair of words $\theta = (v, w)$, both of the same length, over the alphabets $D^{|\Sigma|}$ and $\wp(Prop)$. If the durations in the first (length) component of v are positive, the underlying flow of time is said to be monotonic, otherwise it is said to be weakly monotonic.

Sampled measurement models over the signature ℓ using the real numbers as a duration domain were introduced as timed state sequences (or timed words) by Alur and Dill [1994]. They used an alternate definition where time values $(v_0(\ell))(v_0(\ell) + v_1(\ell))(v_0(\ell) + v_1(\ell) + v_2(\ell)) \ldots$ are used in the sampling sequence. Our definition is consistent with time differences being used. The additivity assumption on the signature makes both definitions equivalent. We will use $\sum_i^j v(m)$ to denote the sum $\Sigma_{l=i+1}^j v_i(m)$, which sums up the measurements over an interval from the differences. Usually, additional constraints are put on infinite behaviours so that time does not converge to a point but diverges to infinity.

An infinite model can be represented as an ω-word if we further assume that such a model is finitely generated, for instance, as an infinite path in a finite transition system. This suggests the model checking question: given a finite automaton and a formula α, do all finite/infinite words which belong to the language accepted by the automaton satisfy α ? The evaluation point is *anchored* at the beginning to v_0, w_0.

From the earlier work of Büchi [1960, 1962] (also Elgot [1961] and Trakhtenbrot [1961] in the case of finite words), it is easy to see that the model checking and satisfiability questions can be reduced to the language inclusion and emptiness problems for automata.

4.1 Measurement logic

While one can define point and interval logics over sampled time behaviours, the absence of the Kripke frame makes the distinctions somewhat arbitrary. Instead one could combine the ideas of both into one system, as Henriksen and Thiagarajan [1999] did using a dynamic temporal logic on sequences. We use a propositional dynamic logic with converse (see the *Dynamic logic* book by Harel, Kozen and Tiuryn [2000] for more details of various dynamic logics).

Our logic generalizes the sampled semantics of *MTL*, given by Alur and Henzinger [1993] and the sampled semantics used for duration logics by Pandya [2002]. The measurement modality $\langle \mu \to \pi \rangle \alpha$ says that there is a behaviour conforming to the program π which satisfies the measurement μ, after which the formula α holds. The modality $\langle -\mu \leftarrow \pi \rangle \alpha$ describes the converse behaviour, going into the past from the present point in the behaviour. Since the full power of regular expressions is used here for the programs, this logic is more expressive than the guarded fragment $\chi GF[\Sigma]$, which has a first order semantics.

Definition 15 (Dynamic measurement logic $\chi DML[\Sigma]$).

$$
\begin{array}{lcl}
\pi & ::= & \lceil p \rceil,\ p \in Prop \mid skip \mid \pi_1 \cup \pi_2 \mid \pi_1 ; \pi_2 \mid \pi^* \\
\mu & ::= & m \sim c,\ m \in \Sigma,\ \ \sim \in \chi,\ c \in D \\
\alpha & ::= & p \mid \neg \alpha \mid \alpha \vee \beta \mid \langle \mu \to \pi \rangle \alpha \mid \langle -\mu \leftarrow \pi \rangle \alpha
\end{array}
$$

Let $\theta = (v, w)$ be a sampled time behaviour. A program π represents a subsequence of the behaviour $(v_i \ldots v_j, w_i \ldots w_j)$ which is specified by the indices i and j. That this subsequence is part of the relation defined by a program π is written as $\theta, [i, j] \models \pi$.

$$\theta, [i,j] \models \lceil p \rceil \text{ iff } \forall z : i \prec z \prec j : \theta(p)[z] = 1$$
$$\theta, [i,j] \models \mathit{skip} \text{ iff } j = i + 1$$
$$\theta, [i,j] \models \pi_1 \cup \pi_2 \text{ iff } \theta[i,j] \models \pi_1 \text{ or } \theta, [i,j] \models \pi_2$$
$$\theta, [i,j] \models \pi_1 ; \pi_2 \text{ iff } \exists z : i \prec z \prec j : \theta, [i,z] \models \pi_1 \text{ and } \theta, [z,j] \models \pi_2$$
$$\theta, [i,j] \models \pi^* \text{ iff } \begin{array}{l} \text{for some } n \geq 0 \text{ and } i = z_0 \prec z_1 \prec \cdots \prec z_n = j : \\ \text{for all } l : 0 \leq l < n : \theta, [z_l, z_{l+1}] \models \pi \end{array}$$

A formula α is evaluated at a time and state v_k, w_k in the model, which is specified by its index k.

$$\theta, k \models p \text{ iff } p \in w_k$$
$$\theta, k \models \langle m \sim c \rightarrow \pi \rangle \alpha \text{ iff } \exists l \succ k : (\textstyle\sum_k^l v(m)) \sim c,\ \theta, [k,l] \models \pi \text{ and } \theta, l \models \alpha$$
$$\theta, k \models \langle -m \sim c \leftarrow \pi \rangle \alpha \text{ iff } \exists j \prec k : (\textstyle\sum_j^k v(m)) \sim c,\ \theta, [j,k] \models \pi \text{ and } \theta, j \models \alpha$$

5 The finite generation of models

Representing sampled-time models by automata is still not trivial. Alur and Dill [1994] defined a timed automaton with a finite number of clocks for this purpose. They also used an extra finite alphabet of letters.

A guarded transition on a letter is a set of clock constraints $x \sim c$ and clock resets which include the target state for convenience (thus $x.s$ stands for resetting the clock x and going to state s).

The finitely many guarded transitions in a finite timed automaton can now describe unboundedly many change points in a finite or infinite behaviour. Hence the automaton provides a finite generation mechanism for sampled time behaviours.

But there is still a catch. The duration domain D can be, and is usually meant to be, infinite. How is a finite automaton supposed to read a letter belonging to an infinite alphabet?

Assume that all the measurement functions in the signature Σ (which we have already assumed to be additive) are order-preserving (as ℓ is) or anti-order-preserving. As we saw in an earlier section, the clocks and constants mentioned in the guards of a finite timed automaton implicitly divide the product duration domain into finitely many "regions" over which the automaton remains in the same state with only the clocks ticking away. This enabled Alur and Dill [1994] to construct a finite region automaton which works on the alphabet of regions (along with the letters) which accepts exactly the "untiming" of the language of the timed automaton. They could decide emptiness of the language of the timed automaton by checking emptiness of the language of the region automaton.

5.1 The formula automaton

What remains is to effect a logic-to-automata translation which reduces the validity of the logic to the emptiness of the language of the automaton. Such a formula automaton is implicit in the work of Büchi, Elgot and Trakhtenbrot. Vardi and Wolper [1994] constructed an explicit formula automaton for temporal logic.

We will be interested in a line of work initiated by Muller, Saoudi and Schupp [1988], who constructed a succinct *alternating* formula automaton for temporal and dynamic logics. (There is an exponential blowup in going from an alternating automaton to an ordinary nondeterministic automaton.) Since the languages accepted by timed automata are not closed under complement, alternating timed automata (which include nondeterministic timed automata and for which the languages accepted are closed under complement) are convenient to use as "formula automata" for logics with duration. They were defined in two papers by Lasota and Walukiewicz [2005] and by Ouaknine and Worrell [2005].

A transition in an alternating timed automaton with clocks X and states Q is a positive boolean combination of guards with clock constraints $x \sim c$ and resets $x.s$. A disjunction means that the automaton chooses one of the disjuncts in its move (as in a nondeterministic automaton), but a conjunction means that the automaton works on *all* conjuncts. We will write $\mathcal{B}^+(Z)$ for the positive boolean combinations over a set Z.

Definition 16. An alternating timed automaton over an alphabet A and a set of clocks X is a tuple $M = (Q, \delta, q_0, F)$, where Q is a finite set of states, $q_0 \in Q, F \subseteq Q$ are the initial state and the set of final states respectively and $\delta : Q \times A \to \mathcal{B}^+(G(X, Q))$ is the guarded transition function.

Defining the run of such an automaton is tedious. We refer to the paper of Ouaknine and Worrell [2005] for the definition. In this paper they construct a 1-clock alternating timed automaton for the future fragment of *Punct-MTL*$[\ell]$. Their construction works only for future formulas and only for finite models. With Pandya [2006], we constructed a 1-clock alternating timed automaton over finite words for a future fragment of *Punct-GIML*$[\ell]$ in which checking lengths is not nested. (This keeps the 1-clock restriction intact.)

Emptiness of the language accepted by an alternating timed automaton restricted to *one* clock was shown to be decidable in the papers of Lasota and Walukiewicz and Ouaknine and Worrell. With two clocks, the problem is known to be undecidable.

Below we construct an alternating timed automaton with the single clock x working on the alphabet $A = \wp(Prop)$ which accepts exactly the finite models of a *pure future* formula α of *Punct-DML*[ℓ]. This means that α does not have past subformulas of the kind $\langle -m \sim c \leftarrow \pi\rangle\beta$. To put it differently, we have a propositional dynamic logic without converse.

We assume all negations in α have been pushed inside to the level of literals. The closure of α is defined, based on the ideas of Fischer and Ladner [1979] for propositional dynamic logic, as used by Ouaknine and Worrell [2005] for *Punct-MTL*[ℓ]. This is used to build the states of the formula automaton.

Definition 17 (Derivatives, closure, formula automaton). For a letter a in A, the a-derivatives $\partial\pi/\partial a$ of a program π are defined inductively:

- The special program $skip^r$ is an a-derivative of $skip$ for any a.
- $\lceil p\rceil$ has itself and $skip^r$ as a-derivatives if $p \in a$ and $false$ otherwise.
- $\pi_1 \cup \pi_2$ has the a-derivatives of π_1 and π_2 as its a-derivatives.
- $\pi_1; \pi_2$ has $\{q_1; \pi_2 \mid q_1 \in \partial\pi_1/\partial a\}$ as its a-derivatives.
  A derivative $skip^r; \pi_2$ is taken to be the same as π_2.
- π^* has $\{skip^r\} \cup \{q; \pi^* \mid q \in \partial\pi/\partial a\}$ as its a-derivatives.

If q is an a-derivative of π, we define the formula $\langle\mu \to q\rangle\alpha$ to be an a-derivative of $\langle\mu \to \pi\rangle\alpha$.

The closure $CL(\alpha)$ of a formula α contains α, a special initial copy α_{init} of α. It is closed under taking subformulas $\gamma = \langle\mu \to \pi\rangle\beta$ with an outermost modality, of a formula already in the closure, and under taking derivatives (of a formula). We also throw in two states $true$ and $false$. It is a standard dynamic logic exercise to check that the closure of a formula is a finite set.

The (1-clock alternating timed) formula automaton of a formula α has $CL(\alpha)$ as its states. α_{init} is the initial state. The $[\mu \to \pi]$ and $\langle\mu \to skip^r\rangle$ formulas and the formula $true$ are the final states. The transition function is given by the clauses below.

$$\begin{array}{rcl}
\delta(\alpha_{init}, a) & = & x.\delta(\alpha, a) \\
\delta(\gamma_1 \vee \gamma_2, a) & = & \delta(\gamma_1, a) \vee \delta(\gamma_2, a) \\
\delta(\gamma_1 \wedge \gamma_2, a) & = & \delta(\gamma_1, a) \wedge \delta(\gamma_2, a) \\
\delta(\langle \mu \to \pi \rangle \beta, a) & = & \bigvee_{b \in A} \bigvee_{q \in \partial\pi/\partial b} (\langle \mu \to q \rangle \beta \wedge x.\delta(a, b)) \\
\delta([\mu \to \pi]\beta, a) & = & \bigwedge_{b \in A} \bigwedge_{q \in \partial\pi/\partial b} ([\mu \to q]\beta \vee x.\delta(a, b)) \\
\delta(\langle \ell \sim c \to \mathit{skip}^r \rangle \beta, a) & = & (x \sim c) \wedge x.\delta(\beta, a) \\
\delta([\ell \sim c \to \mathit{skip}^r]\beta, a) & = & \neg(x \sim c) \vee x.\delta(\beta, a) \\
\delta(\neg p, a) & = & \mathit{true} \text{ for } p \notin a, \ \mathit{false} \text{ for } p \in a \\
\delta(p, a) & = & \mathit{true} \text{ for } p \in a, \ \mathit{false} \text{ for } p \notin a
\end{array}$$

The next theorem is our decidability result. For the reader wondering about its restricted nature, Ouaknine and Worrell showed [2005] that *MTL* (and hence our logic) with both past and future modalities is undecidable; the same is true when infinite models are considered.

Theorem 18. For the future fragment of *Punct-DML*[ℓ] over finite models, model checking and validity are decidable.

Proof. Since the language emptiness and inclusion problems are decidable for one-clock alternating timed automata, it is sufficient to show that the timed language accepted by the formula automaton for α is exactly the finite behaviours where α holds.

Consider a finite behaviour $\theta = (v, w)$ of length n accepted by the formula automaton for α, using the accepting run $u_0 \xrightarrow{v_1, w_1} u_1 \xrightarrow{v_2, w_2} \dots \xrightarrow{v_n, w_n} u_n$. We show for each subformula γ of α and each index i that if the guard $\delta(\gamma, w_{i+1})$ is satisfied in u_i then $\theta, i \models \gamma$. This is shown by structural induction on γ.

The base case, when γ is p or $\neg p$: The guard is satisfied by checking $p \in a$. Correspondingly $\theta, i \models p$ or $\theta, i \models \neg p$.

For the induction step, γ has a modality, say $\gamma = \langle \mu \to \pi \rangle \beta$.

If the guard is satisfied at i, there is a derivative $q \in \partial\pi/\partial a$ such that the guard $\delta(\langle \mu \to q \rangle \beta, b)$ is satisfied at $i+1$. Repeating this argument, we arrive at a position j and a derivative where the guard $\langle \mu \to \mathit{skip}^r \rangle \beta$ is satisfied. Hence $\delta(\beta, w_{j+1})$ is satisfied in u_j and by the induction hypothesis, $\theta, j \models \beta$. At this point, the clock constraint is checked. From the transition function, we see that the clock is not reset going from a modality to its derivative.

Hence the entire execution $\theta[i,j]$ of π satisfies the clock constraint, which agrees with the semantics.

Now we do an inner induction on $k = j - i$ to work out $\theta, [i,j] \models \pi$. We will do this by temporarily forgetting the comparison and arguing that $\theta, k \models \langle true \to q \rangle \beta$ for a suitable derivative q of π.

For the base case, when $k = 0$ we have just seen that $\theta, j \models \langle true \to skip^r \rangle \beta$.

For the inner induction step, consider $i < j$. Suppose the guard $\delta(\gamma, a)$ holds in u_i. Since there is a successor in the behaviour, using the transition function, for some derivative $q \in \partial\pi/\partial a$, the guard $\delta(\langle true \to q \rangle \beta, b)$ is satisfied at $i+1$. By the induction hypothesis $\theta, i+1 \models \langle true \to q \rangle \beta$. By the semantics of the logic, $\theta, i \models \langle true \to \pi \rangle \beta$.

Since we earlier verified that the clock constraint is also satisfied, we finally get that $\theta, i \models \gamma$.

The dual modality can be similarly handled.

By taking $i = 0$ and $\gamma = \alpha$, we have shown that a behaviour accepted by the formula automaton is a model of α. The reverse inclusion follows from the observation that the formula automaton for $\neg\alpha$ is the dual alternating automaton of the one for α and hence accepts the complementary language.

q.e.d.

6 Remarks

There has been no mention of dense or continuous time in the previous sections, since the logics do not even satisfy basic density axioms, and there is no attempt to deal with limits. The philosopher of time will be disappointed to see how little of the structure of time, or of the nature of its metric topology, is needed to develop a usable logic of measurement.

Bojańczyk *et al* [2006] have abstracted the region construction further by considering the marked projection of a data word over D^Σ to C^Σ, where C is a collection of equivalence classes of D. A data automaton works as a two-level process: a letter-to-letter transducer which outputs an equivalence class for each letter of the input (the marked projection of the data word), and a class automaton which works on this information to recognize the language. As might be expected, data logics are an abstraction of logics with duration where the structure of a duration domain is replaced by an equivalence relation.

References.

[1983] J.F. ALLEN. Maintaining knowledge about temporal intervals, *Commun. ACM* 26:11, 832–843.

[1994] R. ALUR AND D. DILL. A theory of timed automata, *TCS* 126, 183–236.

[1993] R. ALUR AND T. HENZINGER. Real-time logics: complexity and expressiveness, *Inf. Comput.* 104:1, 35–77.

[1996] R. ALUR, T. FEDER AND T. HENZINGER. The benefits of relaxing punctuality, *J.ACM* 43:1, 116–146.

[2006] M. BOJAŃCZYK, C. DAVID, A. MUSCHOLL, T. SCHWENTICK AND L. SEGOUFIN. Two-variable logic on words with data, *Proc. LICS* XXI, Seattle, IEEE, 7–16.

[2007] D. BRESOLIN, A. MONTANARI AND G. SCIAVICCO. An optimal decision procedure for right propositional neighborhood logic, *J. Autom. Reas.* 38:1–3, 173–199.

[1960] J.R. BÜCHI. Weak second-order arithmetic and finite automata, *Z. Math. Logik Grundl. Math.* 6, 66–92.

[1962] J.R. BÜCHI. On a decision method in restricted second-order arithmetic, *Proc. 1960 Congr. Logic, Methodology, Philosophy and Science*, Stanford (E. NAGEL, P. SUPPES AND A. TARSKI, eds.), Stanford Univ Press, 1–11.

[1982] J.P. BURGESS. Axioms for tense logic, I: “since” and “until”, II: time periods, *Notre Dame J.FL* 23:4, 367–383.

[1984] J.P. BURGESS. Basic tense logic, in (D.M. GABBAY AND F. GUENTHNER, eds.) *Handbook of philosophical logic* II, Reidel, 89–133.

[1995] B. DUTERTRE. Complete proof systems for first order interval temporal logic, *Proc. LICS* X, San Diego, IEEE, 36–43.

[1961] C.C. ELGOT. Decision problems of finite automata design and related arithmetics, *Trans. AMS* 98, 21–52.

[1979] M.J. FISCHER AND R.E. LADNER. Propositional dynamic logic of regular programs, *J. Comput. Syst. Sci.* 18:2, 194–211.

[1980] D.M. GABBAY, A. PNUELI, S. SHELAH AND J. STAVI. On the temporal basis of fairness, *Proc. POPL* VII, Las Vegas, ACM, 163–173.

[1994] D.M. GABBAY, I. HODKINSON AND M.A. REYNOLDS, *Temporal logic: mathematical foundations and computational aspects* 1, Oxford.

[1995] D.M. GABBAY, C. HOGGER AND J.A. ROBINSON (eds), *Handbook of logic in artificial intelligence and logic programming* IV, Oxford.

[2000] D.M. GABBAY, M.A. REYNOLDS AND M. FINGER, *Temporal logic: mathematical foundations and computational aspects* 2, Oxford.

[1995] A.P. GALTON. Time and change for AI, in D.M. GABBAY, C. HOGGER AND J.A. ROBINSON [1995], 175–240.

[2004] V. GORANKO, A. MONTANARI AND G. SCIAVICCO. A roadmap of interval temporal logics and duration calculi, *J. Appl. Non-Classical Logics* 14:1–2, 9–54.

[1991] J.Y. HALPERN AND Y. SHOHAM. A propositional modal logic of time intervals, *J. ACM* 38:4, 935–962.

[1969] C.L. HAMBLIN. Starting and stopping, *The Monist* 53, 410–425.

[2000] D. HAREL, D. KOZEN AND J. TIURYN. *Dynamic logic*, MIT Press.

[1995] P.J. HAYES. A catalog of temporal theories, Technical Report UIUC-BI-AI-96-01, U. Illinois.

[1999] J.G. Henriksen and P.S. Thiagarajan. Dynamic linear time temporal logic, *Ann. Pure Appl. Logic* 96:1-3, 187–207.

[2005] Y. Hirshfeld and A. Rabinovich. Timer formulas and decidable metric temporal logic, *Inf. Comput.* 198, 148–178.

[2000] I. Hodkinson. Temporal logic and automata, in D.M. Gabbay, M.A. Reynolds and M. Finger [2000], 30–72.

[1968] J.A.W. Kamp. *Tense logic and the theory of linear order*, PhD thesis, UCLA.

[1990] R. Koymans. Specifying real-time properties with metric temporal logic, *Real-time systems* 2:4, 255–299.

[2005] S. Lasota and I. Walukiewicz. Alternating timed automata, *Proc. Fossacs* VIII, Edinburgh (V. Sassone, ed.), *LNCS* 3441, 250–265.

[2000] K. Lodaya. Sharpening the undecidability of interval temporal logic, *Proc. Asian* VI, Penang (J. He and M. Sato, eds.), *LNCS* 1961, 290–298.

[2006] K. Lodaya and P.K. Pandya. A dose of timed logic, in guarded measure, *Proc. Formats* IV, Paris (E. Asarin and P. Bouyer, eds.), *LNCS* 4202, 260–273.

[1992] Z. Manna and A. Pnueli. *The temporal logic of reactive and concurrent systems: specification*, Springer.

[1997] A. Montanari and M. de Rijke. Two-sorted metric temporal logics, *Theoret. Comput. Sci.* 183:2, 187–214.

[1983] B.C. Moszkowski and Z. Manna. Reasoning in interval temporal logic, *Proc. Logics of programs*, Pittsburgh (E.M. Clarke and D. Kozen, eds.), *LNCS* 164, 371–382.

[1988] D.E. Muller, A. Saoudi and P.E. Schupp. Weak alternating automata give a simple explanation of why most temporal and dyhamic logics are decidable in exponential time, *Proc. LICS* III, Edinburgh, IEEE, 422–427.

[2005] J. Ouaknine and J. Worrell. On the decidability of metric temporal logic, *Proc. LICS* XX, Chicago, IEEE, 188–197.

[2002] P.K. Pandya. Interval duration logic: expressiveness and decidability, *Proc. TPTS*, Grenoble (E. Asarin, O. Maler and S. Yovine, eds.), *ENTCS* 65:6, 19 pp.

[1986] R. Rosner and A. Pnueli. A choppy logic, *Proc. LICS* I, Cambridge, IEEE, 306–313.

[1983] R.L. Schwartz, P.M. Melliar-Smith and F.H. Vogt. An interval-based temporal logic, *Proc. Logics of programs*, Pittsburgh (E.M. Clarke and D. Kozen, eds.), *LNCS* 164, 443–457.

[1961] B.A. Trakhtenbrot. Finite automata and the logic of monadic predicates, *Dokl. Akad. Nauk SSSR* 140, 326–329.

[1983] J.F.A.K. van Benthem. *The logic of time*, Reidel.

[1995] J.F.A.K. van Benthem. Temporal logic, in D.M. Gabbay, C. Hogger and J.A. Robinson [1995], 241–350.

[1994] M.Y. Vardi and P. Wolper. Reasoning about infinite computations, *Inf. Comput.* 115:1, 1–37.

[2006] M.Y. Vardi. From Church and Prior to PSL, in (O. Grumberg and H. Veith, eds.) *Symp. 25 years of model checking*, Seattle.

[1990] Y. Venema. Expressiveness and completeness of an interval tense logic, *Notre Dame J.FL* 31:4, 529–547.

[1991] Y. Venema. A modal logic for chopping intervals, *J. Logic Comput.* 1:4, 453–476.

[1982] M.B. Vilain. A system for reasoning about time, *Proc. AAAI* II, Pittsburgh, AAAI Press, 197–201.

[1991] Zhou C., C.A.R. Hoare and A. Ravn. A calculus of durations, *Inform. Proc. Lett.* 40:5, 269–276.

[1993] Zhou C., M.R. Hansen and P. Sestoft. Decidability and undecidability results for duration calculus, *Proc. STACS* X, Würzburg, *LNCS* 665, 58–68.

[2004] Zhou C. and M.R. Hansen. *Duration calculus*, Springer.

∅ q.e.d.