A Dose of Timed Logic, in Guarded Measure

Kamal Lodaya[1] and Paritosh K. Pandya[2]

[1] The Institute of Mathematical Sciences
CIT Campus, Chennai 600113, India
[2] Tata Institute of Fundamental Research
Colaba, Mumbai 400005, India

Abstract. We consider interval measurement logic *IML*, a sublogic of Zhou and Hansen's interval logic, with measurement functions which provide real-valued measurement of some aspect of system behaviour in a given time interval. We interpret *IML* over a variety of time domains (continuous, sampled, integer) and show that it can provide a unified treatment of many diverse temporal logics including duration calculus (DC), interval duration logic (IDL) and metric temporal logic (MTL). We introduce a fragment *GIML* with restricted measurement modalities which subsumes most of the decidable timed logics considered in the literature.

Next, we introduce a guarded first-order logic with measurements *MGF*. As a generalisation of Kamp's theorem, we show that over arbitrary time domains, the measurement logic *GIML* is expressively complete for it. We also show that *MGF* has the 3-variable property.

In addition, we have a preliminary result showing the decidability of a subset of *GIML* when interpreted over timed words.

The importance of reasoning about timed systems has led to considerable research on models and logics for timed behaviours. We consider a slightly more general situation where, in addition to time, we can use other measurement functions as well. For instance, instead of saying "during the last 24 hours, the rainfall was 100 mm," we can say that "the time elapsed for the last 100 mm of rainfall was over 4 months." We can also have measurements of quantities like "mean value" of a proposition within a time interval. Guelev has shown how probabilities might be incorporated into such a framework [Gue00].

Unlike data languages [BPT03], there is no finite state mechanism associated with the measurement functions. Thus we are in the setting of the interval logic with measurements defined by Zhou and Hansen [ZH04].

There exists quite a menagerie of timed and duration logics. In Section 1 below, we review the literature and define our logic $\chi IML[\Sigma]$ over a signature Σ of measurement functions, and parameterised by a set of primitive comparisons χ dependent on Σ. We show that it can provide a unified treatment of many diverse temporal logics including duration calculus (DC), interval duration logic (IDL) and metric temporal logic (MTL).

In Section 2, we consider an enrichment of Kamp's $FO[<]$ with measurements. The undecidability of this logic motivates us to formulate and investigate a

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 260–273, 2006.

fragment $\chi MGF[<, \Sigma]$ with χ-guarded measurement quantifiers. The next two sections show that $\chi GIML$ is expressively complete for χMGF. Kamp's syntactic techniques were used by Venema [Ven91], and we extend these as well as the pebble games of Immerman and Kozen [IK87] in our proofs. As in Kamp's result, we show along the way that χMGF has the three-variable property.

Thus the expressiveness of our logic is reasonably delineated. We would have liked to have established a connection to aperiodic languages [Bac03] but this must remain future work.

We now turn to decidability. We find that *IML* and *GIML* are in general undecidable, but for a set of weak comparisons (which disallow equality tests between measurements and constants), we use a result by Hirshfeld and Rabinovich [HR99] and our expressive completeness to show that *Weak-GIML*$[\ell]$ is decidable for continous time. We also prove by translation into one-clock alternating timed automata [LW05, OW05], decidability over timed words of a sublogic *Punct-FgIML*$[\ell]$ of *GIML*$[\ell]$, which only has nesting-free forward guarding.

1 A Classification of Timed Behaviours and Logics

Timed logics describe the evolution of system behaviour in time. For us, time is a linear order $(T, <)$, and we will further assume that T is a subset of the non-negative reals (which we designate $\Re$) with $<$ the usual ordering. $Intv(T)$, the set of (closed) intervals of T, is $\{[b, e] \in T \times T \mid b \leq e\}$. A time frame $TF = (T, <, d)$ is a subset of the real order $(\Re, <)$ with d giving the absolute value of the distance on the real line between two real numbers, i.e. $d[b, e] = |b - e|$.

Zhou and Hansen have proposed an interesting interval logic [ZH04] where the variables (measurement functions) denote real-valued measurements of system behaviour in a given time interval. Formally we have a signature $\Sigma = \{m_1, \ldots, m_n\}$ of measurement function symbols (of arity 2), and we assume that it contains the distinguished function ℓ which measures the length of the interval. We will often abbreviate the signature $\{\ell\}$ to ℓ.

Zhou and Hansen's logic allows first order real arithmetic over such measurements. In this section, we introduce a restricted version of this logic where a measurement may only be compared with an integer constant. We call this logic **interval measurement logic**, *IML*.

Let $Pvar$ be a finite set of propositional variables. A behaviour of a system over TF is a pair of maps $\theta : (Pvar \to T \to \{0, 1\}) \times (\Sigma \to Intv \to \Re)$, where Σ might depend on $Pvar$. For convenience we write $\theta(p)$ as a boolean function of time and $\theta(m)[b, e]$, for $m \in \Sigma$, as giving the value of the measurement function m on the interval $[b, e]$. Moreover, we require that the measurement ℓ is always interpreted as length of the interval, i.e. $\theta(\ell)[b, e] = d[b, e] = |b - e|$. An interval model is a pair $\theta, [b, e]$.

It is useful to consider several classes of time frames $TF = (T, <, d)$ where $T \subseteq \Re$. In the literature, we find a variety of timed logics which use these different classes as time frames. Some such logics are listed in the next section.

Continuous infinite time. $T = \Re$.

Continuous time finitely variable behaviours. We call a continuous time behaviour θ finitely variable if for any p in $Pvar$, $\theta(p)$ changes only finitely often within a finite interval.

Continuous time prefix behaviours. $T = [0, r]$ for some $r \in \Re$ where $[0, r]$ denotes the set of reals between 0 and r. Let $max(T) = r$ give the maximum time-point upto which the behaviour is captured.

Sampled time infinite behaviours. T has the form $\{r_0, r_1, \ldots\}$, the countably infinite set of sampling points where $r_0 = 0$ and $r_0, r_1, \ldots$ forms an unbounded increasing sequence within $\Re$. These behaviours are also called timed ω-words.

Sampled time prefix behaviours. T has the form $\{r_0, r_1, \ldots, r_k\}$, the finite set of sampling points where $r_0 = 0$ and $r_i \in \Re$. Also $r_i < r_{i+1}$. Let $max(T) = r_k$. These behaviours are also called finite timed words.

Discrete time. This is a subclass of sampled time behaviour (infinite or prefix) where all sampling points have integer values.

1.1 Interval Measurement Logic

The formulae of interval measurement logic $\chi IML[\Sigma]$ are parameterised by a set χ of atomic measurement comparisons. For concreteness, let us fix $Punct(\Sigma)$ to be the countable set of comparisons $m \sim c$, for all $m \in \Sigma$, $\sim$ in $\{<, =, >\}$ and c in $\mathbf{Z}$, the set of integers. Since punctuality is a strong requirement [AFH96], we also define $Weak(\Sigma)$ to be the subset of weak comparisons made only using the $<$ symbol, and $Test(\Sigma)$ to be the set of comparisons of the form $m = 0$.

Boolean combinations of the propositional variables $Pvar$ and $0, 1$ (denoting *false* and *true* respectively) are called propositions, $Prop$. Let P, Q range over propositions, $m \sim c$ over comparisons from a set χ and D_1, D_2 over formulae. The formulae of $\chi IML[\Sigma]$ have the syntax

$$\lceil P \rceil^0 \;|\; \lceil P \rceil \;|\; m \sim c \;|\; D_1 {}^\frown D_2 \;|\; D_1 \overset{\frown}{+} D_2 \;|\; D_1 \overset{\frown}{-} D_2 \;|\; D_1 \wedge D_2 \;|\; \neg D$$

When we write $IML[\Sigma]$ we mean that χ is the full set of comparisons $Punct(\Sigma)$.

Semantics of IML. For a proposition P and time point t, $\theta, t \models P$ is defined inductively as usual. Let $\theta, [b, e] \models D$ denote that the formula $D \in IML$ evaluates to true in the behaviour θ at interval $[b, e] \in Intv(\theta)$. Omitting the boolean cases, this is defined as follows.

$\theta, [b, e] \models \lceil P \rceil^0$ iff $b = e$ and $\theta, b \models P$
$\theta, [b, e] \models \lceil P \rceil$ iff $b < e$ and for all $t :\; b < t < e.\; \theta, t \models P$
$\theta, [b, e] \models m \sim c$ iff $\theta(m)[b, e] \sim c$
$\theta, [b, e] \models D_1 {}^\frown D_2$ iff for some $z : b \leq z \leq e.\; \theta, [b, z] \models D_1$ and $\theta, [z, e] \models D_2$
$\theta, [b, e] \models D_1 \overset{\frown}{+} D_2$ iff for some $z : e \leq z.\; \theta, [b, z] \models D_1$ and $\theta, [e, z] \models D_2$
$\theta, [b, e] \models D_1 \overset{\frown}{-} D_2$ iff for some $z : z \leq b.\; \theta, [z, e] \models D_1$ and $\theta, [z, b] \models D_2$

Derived operators. Note that $\lceil 1 \rceil^0$ holds for all point intervals whereas $\lceil 1 \rceil$ holds for all extended intervals. The formula $\lceil\lceil P \rceil \stackrel{\text{def}}{=} \lceil P \rceil^0 \frown \lceil P \rceil$ states that P must hold invariantly over the interval, except possibly at the last point.

- $\Diamond D \stackrel{\text{def}}{=} true \frown D \frown true$ holds provided D holds for some subinterval.
- $\Box D \stackrel{\text{def}}{=} \neg\Diamond\neg D$ holds provided D holds for all subintervals.
- $\overrightarrow{\Diamond}\, D \stackrel{\text{def}}{=} D \stackrel{\frown}{+} true$ holds provided some forward extension of the interval satisfies D. Symmetrically $\overleftarrow{\Diamond}\, D \stackrel{\text{def}}{=} true \stackrel{\frown}{-} D$.

Validity. As usual D is valid iff for all behaviours θ, $\theta \models D$, where

- For prefix behaviours, $\theta \models D$ iff $\theta, [0, max(\theta)] \models D$
- For infinite behaviours, $\theta \models D$ iff $\theta, [0, e] \models D$ for all $e \in dom(\theta)$.

Example 1. The formula $\Box(\lceil\lceil P \rceil \Rightarrow \ell \leq c)$ states that P can be continuously true for at most c time units.

Various sublogics of *IML* have appeared in the literature. We use different signatures to relate our work to a few of these. (The original versions of some of these logics do not include the modalities $D_1 \stackrel{\frown}{+} D_2$ and $D_1 \stackrel{\frown}{-} D_2$ which were introduced by Venema [Ven90].)

Duration calculi. Let the signature $Duration(Pvar) = \{\ell\} \cup \{\int P \mid P \in Prop\}$. The term $\int P$ is interpreted to measure the accumulated amount of time for which proposition P is true in an interval. Thus, we obtain the logic *Punct-IML*[*Duration*(*Pvar*)]. This logic is called duration calculus, DC, when interpreted over continuous time finitely variable models [ZH04]; interval duration logic, IDL, when interpreted over sampled time prefix models [Pan02]; and DDC when interpreted over integer time prefix models.

Mean value calculus. Let the signature $Mean(Pvar) = \{\ell\} \cup \{\overline{P} \mid P \in Prop\}$. The term $\overline{P}$ is interpreted to measure the mean value of proposition P in an interval $[b, e]$. The logic *Punct-IML*[*Mean*(*Pvar*)] is the mean value calculus, MVC, interpreted over continuous time finitely variable models [ZL94].

CDT. Consider a signature Σ of measurement functions *without* ℓ. If we only allow comparisons with zero—that is, χ is $Test(\Sigma)$—effectively we are restricting from real-valued measurements to boolean-valued ones. Such a measurement function is nothing but an atomic proposition (such as “did it rain?”) evaluated at every interval. This is an idea which has been long studied by philosophers of time. The corresponding logic *Test-IML*[Σ] was called CDT [Ven91], as the modalities $\frown$, $\stackrel{\frown}{-}$, $\stackrel{\frown}{+}$ are named C, D and T respectively.

Interval length logic. On the other hand, we can consider the signature $\{\ell\}$ without any other measurement functions. The logic *Punct-IML*[ℓ] is called interval length logic. As in most real-time logics, it only includes the measurement of time distance using the distinguished function ℓ.

Interval temporal logic. Finally, the trivial logic *IML*[$\emptyset$] with the empty signature is called interval temporal logic, ITL. This logic has been studied over all the classes of models discussed above.

The discrete time logic DDC has been shown to be decidable using an automata-theoretic decision procedure [Pan02]. The general situation is bleaker.

Proposition 1. *The logic Punct-IML*[ℓ] *is undecidable for continuous, finitely variable and sampled behaviours whether infinite or prefix. Test-IML*[Σ] *is undecidable for infinite time.*

Proof. As in the undecidability proof for DC [ZH04], for each 2-counter machine M we can define a formula $D(M)$ of *Punct-IML*[ℓ] which is satisfiable iff M has a halting run. The nonhalting problem is encoded using a very narrow subset of CDT, with unary modalities definable from $\frown$, in [Lod00].

1.2 Guarded Measurement

Faced with the strong undecidability results described above, we restrict the logic by permitting only *guarded* use of measurement formulae. A **χ-guarded modality** has the syntax

$$G \rightarrow D \;\mid\; G \leftarrow D,$$

where the guard G is a boolean formula over the set of comparisons χ. The meaning of guarded modalities is as follows:

$$\theta, [b,e] \models G \rightarrow D \;\text{ iff }\; b = e \text{ and } \text{ for some } \; z : b \leq z.\; \theta, [b,z] \models G \wedge D$$
$$\theta, [b,e] \models G \leftarrow D \;\text{ iff }\; b = e \text{ and } \text{ for some } \; z : z \leq b.\; \theta, [z,b] \models G \wedge D$$

Formally, $G \rightarrow D \;\stackrel{\text{def}}{=}\; \lceil 1 \rceil^0 \wedge \overrightarrow{\Diamond}\,(G \wedge D)$ and $G \leftarrow D \;\stackrel{\text{def}}{=}\; \lceil 1 \rceil^0 \wedge \overleftarrow{\Diamond}\,(G \wedge D)$.

Example 2. The formula $\Box(\neg(\ell \leq c) \rightarrow \neg[\lceil P \rceil])$ is *Weak*-guarded and states that P can be continuously true only for at most c time units.

χ*GIML*[Σ] is the sublogic of *IML*[Σ] where measurements only appear in χ-guards. Thus, *Punct-GIML*[Σ] guards use boolean combinations of comparisons from *Punct*(Σ). If only forward (resp. backward) measurement guards are used, we call the logic *FGIML* (resp. *BGIML*). If in the modality $G \rightarrow D$ of *FGIML*, we do not allow guarded modalities in D, we get a logic with nesting-free forward guarding, which we denote *Punct-FgIML*. Guarded modalities exist in the literature, though not in direct fashion.

Relative distance. A subset of interval duration logic *IML*[*Duration*($Pvar$)] where measurements only occur within the guard G of a modality $P \rightsquigarrow G$ (originally due to [Wil94]) has been called LIDL [Pan02]. This logic can be encoded in the backwards guarded logic *BGIML*[*Duration*($Pvar$)] by encoding the LIDL formula $P \rightsquigarrow G$ as the *BGIML* formula $G \leftarrow \lceil P \rceil^0 \frown \lceil \neg P \rceil$. All the other constructs of LIDL are already available in *BGIML*.

Metric temporal logic. The logic MTL [Koy90] can be encoded in *Punct-FGIML*[ℓ] as follows (see [Pan96] for details). For every MTL formula ϕ we define a translation $\alpha(\phi)$: Let $\alpha(p) = \lceil p \rceil^0 \frown true$. Let $BP(D) \;\stackrel{\text{def}}{=}$

$(\lceil 1 \rceil^0 \wedge \overrightarrow{\Diamond} D) \frown true$. Then, the constrained until modality of MTL is encoded as follows.

$$\alpha(\phi\, \mathcal{U}_I\, \psi) = (I(\ell) \rightarrow \neg(true \frown BP(\alpha(\phi) \frown true)) \frown BP(\alpha(\psi))) \frown true$$

Here $I(\ell)$ is the constraint corresponding to the interval I, e.g. for $[3, 5)$ we get $\ell \geq 3 \wedge \ell < 5$. It can be shown that for all θ, b, e, ϕ we have $\theta, b \models_{mtl} \phi$ iff $\theta, [b, e] \models_{iml} \alpha(\phi)$. By a variation of this construction, we can model MTL with both past and future modalities in *Punct-GIML*$[\ell]$.

Guarding in first order logic has been an important tool for obtaining decidability. Unfortunately, we can show that in the presence of punctual measurements guarding does not guarantee decidability. MTL with future operators is undecidable over continuous infinite time and MTL with past and future operators is undecidable over sampled prefix time [OW05] and the second author and Vijay Suman have recently shown that LIDL is undecidable over sampled time, prefix or infinite. (However, LIDL with only length measurements is decidable over sampled prefix time [Pan02].) All these logics can be encoded within fragments of *GIML* giving the following results.

Proposition 2. *1. The logic Punct-FGIML*$[\ell]$ *is undecidable for continuous infinite time.*
2. The logic Punct-GIML$[\ell]$ *is undecidable for sampled time.*
3. The logic Punct-BGIML$[Duration(Pvar)]$ *is undecidable for sampled time.*

2 First Order Logics with Measurement

In place of interval measurement logic, we can specify a behaviour θ using the first order logic with measurement $MFO[\widehat{\Sigma}]$. This is the first order logic with equality over the signature $\widehat{\Sigma} = (Pvar, \{<\}, \Sigma)$ where each $p \in Pvar$ denotes a monadic predicate.

Example 3. The formula $\forall x, y.\ x < y \wedge (\forall z.\ x < z < y \Rightarrow P(z)) \Rightarrow \ell(x, y) \leq c$ states that P cannot be true continuously for more than c time units.

We can associate a classical first order structure $\underline{\theta}$ interpreting $\widehat{\Sigma}$ with a given behaviour θ. The domain of $\underline{\theta}$ is T with linear order $<$. For each $p \in Pvar$ there is a monadic predicate $p(x)$ which is interpreted as the set $\theta(p)$. The functions $m \in \Sigma$ are interpreted as $\theta(m)$. The semantics of $MFO[\widehat{\Sigma}]$ is given as usual and omitted here.

Proposition 3. *There is a bijection* $(\theta, \underline{\theta})$ *between the* $IML[\Sigma]$ *behaviours and the first-order structures interpreting* $\widehat{\Sigma}$.

While the monadic theory of linear order $MonFO[<]$ is decidable [LL66], introduction of even the basic measurement $\Sigma = \{\ell\}$ makes the logic $MFO[\widehat{\Sigma}]$

undecidable [ZH04]. Hence we resort to a notion of guarded use of measurements. Let $\chi MGF[\Sigma]$ be the measurement-guarded fragment of $MFO[\widehat{\Sigma}]$ which extends $MonFO[<]$ 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 over the signature Σ. Thus the measurement terms appear in a very restricted context.

We now translate our interval measurement logics into measurement guarded first order logics.

The notation $\phi(x,y)$ indicates a formula with at most two free variables x and y. We will use notation such as $FO(x,y)$ to indicate a logic with formulas with at most two free variables x and y. Superscripts as in FO^k designate k-variable fragments of a logic (now including bound as well as free variables).

$$
\begin{array}{lcl}
ST_z(\lceil P \rceil^0)(x,y) & \stackrel{\text{def}}{=} & x = y \wedge P(x) \\
ST_z(\lceil P \rceil)(x,y) & \stackrel{\text{def}}{=} & x < y \wedge \forall z (x < z < y \Rightarrow P(z)) \\
ST_z(G \rightarrow D)(x,y) & \stackrel{\text{def}}{=} & x = y \wedge \exists z (ST(G)(y,z) \wedge y \leq z \wedge ST_x(D)(y,z)) \\
ST_z(G \leftarrow D)(x,y) & \stackrel{\text{def}}{=} & x = y \wedge \exists z (ST(G)(z,x) \wedge z \leq x \wedge ST_y(D)(z,x)) \\
ST_z(D_1 \frown D_2)(x,y) & \stackrel{\text{def}}{=} & \exists z (x \leq z \leq y \wedge ST_y(D_1)(x,z) \wedge ST_x(D_2)(z,y)) \\
ST_z(D_1 \stackrel{\frown}{-} D_2)(x,y) & \stackrel{\text{def}}{=} & \exists z (z \leq x \wedge ST_y(D_1)(z,x) \wedge ST_x(D_2)(z,y)) \\
ST_z(D_1 \stackrel{\frown}{+} D_2)(x,y) & \stackrel{\text{def}}{=} & \exists z (y \leq z \wedge ST_y(D_1)(x,z) \wedge ST_x(D_2)(y,z))
\end{array}
$$

The translation of guards is obvious: $ST(m \sim c)(x,y) = m(x,y) \sim c$. The translation uses the standard trick of reusing variables. Thus $ST_z(D)(x,y)$ produces a $MGF^3(x,y)$ formula using at most the variables $\{x,y,z\}$.

Proposition 4. *There is a standard translation from $\chi GIML[\Sigma]$ to $\chi MGF^3[\Sigma]$ which has the property that $\theta,[b,e] \models D$ iff $\underline{\theta} \models ST_z(D)[b/x, e/y]$.*

3 Expressive Completeness of *GIML* for *MGF*³

Without loss of generality we assume the logic χMGF^3 consists of formulae with variables x_1, x_2, x_3. In this section, following the proof of Kamp's theorem [Kamp68] as used by Venema [Ven91], we show that the measurement logic $\chi GIML$ has the same expressive power as χMGF^3.

The first lemma is routine [GO]. Let $L_{i,j}$, $i \neq j$, $i,j \in \{1,2,3\}$, be the subset of $\chi MGF^3(x_i,x_j)$ consisting of boolean combinations of quantifier-free formulas of MGF^3 and quantified MGF^3 formulas with one free variable in $\{x_i,x_j\}$. $L_{i,j}$ is the same as $L_{j,i}$.

Lemma 1. *Any MGF^3 formula is equivalent to a boolean combination of formulae from $L_{1,2} \cup L_{2,3} \cup L_{3,1}$.*

Now we translate $L_{i,j}$ to *GIML*. Following [Ven91], we use a forward translation $\alpha^+ : L_{i,j} \rightarrow GIML$ and a backward translation $\alpha^- : L_{i,j} \rightarrow GIML$. The boolean cases are routine. We assume the measurement functions are symmetric.

$$
\begin{array}{ll}
\alpha^+(x = x) = \mathit{true} & \alpha^-(x = x) = \mathit{true} \\
\alpha^+(x_i = x_j) = \lceil 1 \rceil^0 & \alpha^-(x_i = x_j) = \lceil 1 \rceil^0 \\
\alpha^+(x_i < x_j) = \neg \lceil 1 \rceil^0 & \alpha^-(x_i < x_j) = \mathit{false} \\
\alpha^+(x_j < x_i) = \mathit{false} & \alpha^-(x_j < x_i) = \neg \lceil 1 \rceil^0 \\
\alpha^+(x < x) = \mathit{false} & \alpha^-(x < x) = \mathit{false} \\
\alpha^+(P(x_i)) = \lceil P \rceil^0 \frown \mathit{true} & \alpha^-(P(x_i)) = \mathit{true} \frown \lceil P \rceil^0 \\
\alpha^+(P(x_j)) = \mathit{true} \frown \lceil P \rceil^0 & \alpha^-(P(x_j)) = \lceil P \rceil^0 \frown \mathit{true} \\
\alpha^+(m(x, y) \sim c) = m \sim c & \alpha^-(m(x, y) \sim c) = m \sim c \\
\alpha^+(\phi_1(x_i, x_j) \wedge \phi_2(x_i, x_j)) & \alpha^-(\phi_1(x_i, x_j) \wedge \phi_2(x_i, x_j)) \\
\quad = \alpha^+(\phi_1(x_i, x_j) \wedge \alpha^+(\phi_2(x_i, x_j)) & \quad = \alpha^-(\phi_1(x_i, x_j)) \wedge \alpha^-(\phi_2(x_i, x_j)) \\
\alpha^+(\neg\phi(x_i, x_j)) = \neg\alpha^+(\phi(x_i, x_j)) & \alpha^-(\neg\phi(x_i, x_j)) = \neg\alpha^-(\phi(x_i, x_j))
\end{array}
$$

The translation of a quantifier uses the fact that the $\frown$, $\overset{\frown}{-}$, $\overset{\frown}{+}$ modalities cover all cases in which a third time point can be oriented with respect to two points.

$$
\begin{array}{l}
\alpha^+(\exists x_k.\ \phi_1(x_i, x_k) \wedge \phi_2(x_k, x_j)) = \alpha^+(\phi_1(x_i, x_k)) \frown \alpha^+(\phi_2(x_k, x_j)) \quad \vee \\
\quad \alpha^+(\phi_1(x_i, x_k)) \overset{\frown}{+} \alpha^-(\phi_2(x_k, x_j)) \ \vee \ \alpha^-(\phi_1(x_i, x_k)) \overset{\frown}{-} \alpha^+(\phi_2(x_k, x_j)) \\
\alpha^-(\exists x_k.\ \phi_1(x_i, x_k) \wedge \phi_2(x_k, x_j)) = \alpha^-(\phi_2(x_k, x_j)) \frown \alpha^-(\phi_1(x_i, x_k)) \quad \vee \\
\quad \alpha^-(\phi_2(x_k, x_j)) \overset{\frown}{+} \alpha^+(\phi_1(x_i, x_k)) \ \vee \ \alpha^+(\phi_2(x_k, x_j)) \overset{\frown}{-} \alpha^-(\phi_1(x_i, x_k))
\end{array}
$$

The translation of a measurement guarded formula uses the forward and backward guarded modalities to cover the way the quantified variable is oriented with respect to the free variable of the formula.

$$
\begin{array}{l}
\alpha^+(\exists x_k.\ G(x_i, x_k) \wedge \zeta(x_i, x_k)) = \\
\quad [\alpha^+(G(x_i, x_k)) \to \alpha^+(\zeta(x_i, x_k)) \ \vee \ \alpha^-(G(x_i, x_k)) \leftarrow \alpha^-(\zeta(x_i, x_k))] \frown \mathit{true} \\
\alpha^-(\exists x_k.\ G(x_i, x_k) \wedge \zeta(x_i, x_k)) = \\
\quad \mathit{true} \frown [\alpha^+(G(x_i, x_k)) \to \alpha^+(\zeta(x_i, x_k)) \ \vee \ \alpha^-(G(x_i, x_k)) \leftarrow \alpha^-(\zeta(x_i, x_k))] \\
\alpha^+(\exists x_k.\ G(x_j, x_k) \wedge \zeta(x_j, x_k)) = \\
\quad \mathit{true} \frown [\alpha^+(G(x_j, x_k)) \to \alpha^+(\zeta(x_j, x_k)) \ \vee \ \alpha^-(G(x_j, x_k)) \leftarrow \alpha^-(\zeta(x_j, x_k))] \\
\alpha^-(\exists x_k.\ G(x_j, x_k) \wedge \zeta(x_j, x_k)) = \\
\quad [\alpha^+(G(x_j, x_k)) \to \alpha^+(\zeta(x_j, x_k)) \ \vee \ \alpha^-(G(x_j, x_k)) \leftarrow \alpha^-(\zeta(x_j, x_k))] \frown \mathit{true}
\end{array}
$$

By a careful case analysis over the syntax of $L_{i,j}$, we can show that the translations α^+ and α^- preserve the semantics in the expected way.

Lemma 2. *For all θ and all $[b, e] \in \mathit{Intv}(\theta)$,*
$\theta, [b, e] \models \alpha^+(\zeta(x_i, x_j))$ *iff* $\underline{\theta} \models \zeta[b/x_i, e/x_j]$ *and*
$\theta, [b, e] \models \alpha^-(\zeta(x_i, x_j))$ *iff* $\underline{\theta} \models \zeta[e/x_i, b/x_j]$.

By combining Lemmas 1 and 2, and observing that the translation above can be parameterised by the set of guards χ, we get the following theorem.

Theorem 1. *The logic $\chi GIML[\Sigma]$ is expressively complete for the three-variable measurement guarded fragment with two free variables $\chi MGF^3[\Sigma](x, y)$.*

A referee reminded us that our proof works for an even larger family of logics: $\chi IML[\Sigma]$ is expressively complete for $\chi FO^3[\Sigma](x, y)$.

4 Games and the 3-Variable Property

Next we would like to show that the full logic χMGF has the three-variable property, that is, χMGF^3 is expressively equivalent to it. To do this, we set up Ehrenfeucht-Fraïssé games for the k-variable guarded fragments, which are an extension of the k-pebble games for FO^k [IK87].

Our game is played for n rounds by two players, Spoiler and Duplicator, on a board consisting of a pair of structures $\mathcal{A}$ and $\mathcal{B}$. Spoiler is trying to distinguish the two structures, Duplicator to hide the distinctions. Each player uses k pebbles for the syntactic restriction to k variables and a measuring tape and meters to check lengths and measurement values. These devices work in integer units.

A k-configuration of consists of the positions of the k pebbles on each structure, which we represent by a pair of partial functions (which are defined where the corresponding pebbles are on the board) $\alpha : \{1..k\} \rightarrow \mathcal{A}$ and $\beta : \{1..k\} \rightarrow \mathcal{B}$. The k-pebble n-round game on structures $\mathcal{A}, \mathcal{B}$ with k-configurations α, β is denoted $G_n^k(\mathcal{A}, \alpha, \mathcal{B}, \beta)$.

Two configurations are said to be order isomorphic if the sequence of pebble positions, seen as linear orders, are order isomorphic. More precisely, Spoiler's pebble i is on the board on one structure if and only if Duplicator's pebble i is present on the other structure, and for each pair of pebbles i, j on the board, both structures satisfy the same formulas from the set $\{i < j, i = j, i > j\}$. By linearity, they will satisfy exactly one formula from this set. Two configurations are said to be χ-measurement isomorphic if they are order isomorphic and for each pair of pebbles i, j on the board, both structures satisfy the same measurement formulas from the set χ.

If α, β are not order isomorphic, Spoiler wins $G_n^k(\mathcal{A}, \alpha, \mathcal{B}, \beta)$ immediately (after 0 rounds). Each round has one of two kinds of moves.

In a pebble move, Spoiler can place his pebble i on an element of one of the structures. Duplicator responds by placing her pebble i on an element of the other structure. After the move, if the resulting configurations α', β' are not order isomorphic, Spoiler wins the game.

In a measuring move, Spoiler removes all pebbles but one, say pebble i (we call this the free pebble of this move), then places another pebble j, using the measuring tape and meters to set its length and other measurement functions to some desired value. Duplicator has to follow suit on the other structure: she removes all pebbles except pebble i, then places her pebble j using the measuring tape and meters. After the move, if the resulting configurations α', β' are not measurement isomorphic, Spoiler wins the game.

If Spoiler has not won the game after any of the n rounds, Duplicator wins $G_n^k(\mathcal{A}, \alpha, \mathcal{B}, \beta)$.

Following [Imm98], we now relate our games to logical types. The proof relies on the fact that the set of measurement formulas $\{m(i, j) \sim c \mid c \in Z\}$ satisfied by a configuration is logically equivalent to a finite conjunction of such formulas, since each value is either the point c or inside an open interval $(c, c + 1)$.

Theorem 2 (E-F characterization). *Given two time frames $\mathcal{A}, \mathcal{B}$ and a k-configuration α_0, β_0 over them, Duplicator wins an n-move game $G_n^k(\mathcal{A}, \alpha_0, \mathcal{B}, \beta_0)$ if and only if the configurations $(\mathcal{A}, \alpha_0)$ and $(\mathcal{B}, \beta_0)$ are indistinguishable by a χMGF^k formula of quantifier depth n.*

We now show that for time domains, three variables suffice to express all *MGF* properties. The proof closely follows the (admittedly tricky) one of [Imm98, Theorem 6.32], which combines winning strategies from simpler games. The measuring move does not yield any difficulty since it always reduces the board to 2-configurations for which a winning strategy exists by supposition.

Theorem 3 (3-variable property). *Every χMGF formula is equivalent to an χMGF^3 formula over time domains.*

Putting together the 3-variable property with the expressive completeness result of the previous section, we get a proper generalization of Kamp's theorem. Venema has shown that $\chi FO[\Sigma]$ does not have the 3-variable property [Ven90], so the result cannot be extended to full first order logic with measurements.

Corollary 1. *The logic $\chi GIML[\Sigma]$ is expressively complete for $\chi MGF[\Sigma]$.*

Hirshfeld and Rabinovich conjectured that there is no finite expressively complete temporal logic for a logic L_2 which subsumes *Weak-MGF*$[\ell]$ by having a more generous set of comparisons [HR99]. We observe that since our logic uses countably many constants c, it is not finite according to their definition.

5 Decidability

One of the main motivations for considering the sublogic *GIML* of *IML* is the hope of getting reasonable decidability properties. In this section we restrict our attention to only the measurement of length, that is, the signature $\Sigma = \{\ell\}$.

Theorem 4. *Over sampled as well as continuous time infinite models, the logic Punct-MGF$[\ell]$ is undecidable and the logics Weak-MGF$[\ell]$ and Weak-GIML$[\ell]$ are decidable.*

Proof. It is shown in the previous section that *Punct-GIML*$[\ell]$ and *Weak-GIML*$[\ell]$ are expressively equivalent to *Punct-MGF*$[\ell]$ and *Weak-MGF*$[\ell]$ and can be translated to these logics. The undecidability of *Punct-MGF*$[\ell]$ for sampled and continuous time infinite models follows from that of *FGIML*$[\ell]$ (Proposition 2). *Weak-GIML*$[\ell]$ is decidable since *Weak-MGF*$[\ell]$ is subsumed by the logic L_2, which was shown decidable over continuous time infinite models [HR99].

The exact decidability border between *Punct-GIML*$[\ell]$ and *Weak-GIML*$[\ell]$ is not clear. As a preliminary result, we show that for sampled time prefix models (i.e. timed words), the logic *Punct-FgIML*$[\ell]$, the nesting-free subset of *FGIML*$[\ell]$, is decidable by reduction to alternating timed automata.

5.1 Alternating Timed Automata

Let C be a finite set of clock variables (more briefly, clocks). A clock valuation v is a function $C \to \Re$. The clock valuation $v + t$ is defined by adding t to each clock value, and the valuation $v[r := 0]$, for $r \subseteq C$, is defined by resetting all the clocks in r to zero. The initial valuation v_0 has all clocks set to zero.

A clock constraint is a boolean combination of conditions $x \sim c$ where x is a clock. Let $Cons(C)$ be the constraints over clocks in C.

Definition 1 (Lasota and Walukiewicz, Ouaknine and Worrell). *An **alternating timed automaton** over the alphabet A and clocks C is a tuple $M = (Q, \delta, I, F)$, where Q is a finite set of states, $I, F \subseteq Q$ are the initial and final states respectively, and $\delta : Q \times A \times Cons(C) \to \mathcal{B}^+(Q \times \wp(C))$ a finite partial transition function, satisfying the partition condition: for every $q \in Q$ and $a \in A$, the set of constraints $\{[\sigma] \mid \delta(q, a, \sigma)\ is\ defined\}$ partitions the set of clock valuations $C \to \Re$. ($\mathcal{B}^+(X)$ is positive boolean formulas over X.)*

A timed word over A is a sequence over $A \times \Re$. The second (time) component gives the time elapsed between reading the previous letter and the current one.

The acceptance game $G_{M,w}$ between two players Pathfinder and Automaton is defined as follows. Automaton's objective is to accept $w = (a_1, t_1) \dots (a_n, t_n)$, Pathfinder's is to reject. A play starts at the configuration (q_0, v_0) and proceeds for $n = |w|$ rounds. Suppose the configuration reached at the end of the i'th round is (q_i, v_i). Let σ be the unique constraint satisfied by the valuation $v_i + t_{i+1}$ and $\delta(q_i, a_{i+1}, \sigma)$ is the formula b. Now there are three cases: if b is a conjunction, Pathfinder chooses one of the conjuncts; if b is a disjunction, Automaton chooses one of the disjuncts; and if $b = q$, the round ends with $q_{i+1} = q$ and $v_{i+1} = (v_i + t_{i+1})[x := 0, x \in \rho(q_{i+1})]$. Automaton wins the game if q_n is a final state, otherwise Pathfinder wins.

The automaton M accepts the timed word w if and only if Automaton has a winning strategy in the game $G_{M,w}$.

The languages accepted by ATA are closed under boolean operations. The papers [LW05, OW05] showed that the emptiness problem for ATA with one clock is decidable. It follows from [AD94] that the problem is undecidable for ATA with two clocks. It is known [LW05] that ATA, even with one clock, are incomparable in expressive power to the usual nondeterministic timed automata of Alur and Dill [AD94].

In order to accept timed languages defined by formulas, automata have to work over models of these formulas. Typically an alphabet 2^{Pvar} is used.

5.2 Decidability of the Nesting-Free Logic *Punct-FgIML*[ℓ]

Recall that *Punct-FgIML*[ℓ] is the subset of *Punct-GIML*[ℓ] with only forward guarded modalities $G \to D$ and where no guarded modality occurs in D. Moreover, we will confine ourselves to sampled time prefix models (finite timed words). The usual Alur-Dill timed automata [AD94], as well as the ATA introduced in the previous section, are recognisers for such timed words. They have decidable emptiness checking. We give an automata-theoretic decision procedure for *Punct-FgIML*$[\ell]$ by reduction to emptiness of ATA. This logic can already express properties not recognised by any timed automaton.

Example 4. Consider behaviour over a single propositional variable. The following property says that there are no pairs of positions exactly one time unit apart: $\neg(true \frown (\ell = 1 \rightarrow true) \frown true)$.

Theorem 5. *For each $D \in$ Punct-FgIML$[\ell]$ over Pvar, we can construct ATA $A(D)$ over alphabet 2^{Pvar} with a single clock x such that $\theta \models D$ iff $\theta \in L(A(D))$.*

Corollary 2. *Punct-FgIML$[\ell]$ is decidable for sampled time prefix models.*

Proof (of Theorem 5). We give the construction of $A(D)$ inductively as follows. Some specific features of our automata are first outlined: These automata have a unique starting state which is never accepting. Our automata never accept the empty word. The automata may also have two distinguished states $\top, \bot$ where $\top$ is accepting and $\bot$ is rejecting. We assume that $\delta(\top, P, true) = \top$ and $\delta(\bot, P, true)$ for all P. The symbol x denotes that the unique clock x is reset and $\overline{x}$ that it is not reset.

(i) In the base case we have $D \in ITL$ without any measurements. Then, we can straightforwardly construct a DFA $A(D)$ recognising the models of D. Note that the models of D satisfiable at an interval $[b, e]$ with $b = e$ are accepted by $A(D)$ with a transition from an initial state to a final state.

(ii) Next, we construct one-clock ATA for $GQ = G \rightarrow D$. By the nesting-free property, D is a pure ITL formula without measurements. Let $A(D) = (Q, \delta, q_0, F)$ be the DFA for D. Then $A(GQ) = (Q \cup \{\top, q_0'\}, \delta', q_0', \{\top\})$. The transition relation δ' is defined as follows.

- Let $\delta(q_0, P) = q$. If $G[0/\ell]$ evaluates to $true$ and $q \in F$, then $\delta'(q_0', P, true) = (\top, \overline{x})$. Otherwise, $\delta'(q_0', P, true) = (q, x)$.
- Let $\delta(q, P) = q'$. If $q' \notin F$, then $\delta'(q, P, true) = (q', \overline{x})$. Otherwise, $q' \in F$ and we have $\delta'(q, P, G[x/\ell]) = (\top, \overline{x})$ and $\delta'(q, P, \neg G[x/\ell]) = (q', \overline{x})$.

Claim: $A(GQ)$ accepts all nonempty words θ such that $\theta, [0, 0] \models GQ \frown true$.

To prove the claim, by the property mentioned in *(i)*, the models of length 0 are obtained by the immediate transitions to the final state. For the other models, the automaton moves off from the initial state (the clock condition for these initial transitions is $true$) and reaches the final state after checking the guard. Since by our convention $\delta(\top, P, true) = \top$, the automaton will continue accepting an extension of a GQ behaviour.

$A(GQ)$ is a deterministic one-clock timed automaton (without alternation). Let $\neg A(GQ)$ denote the complement of $A(GQ)$ where all $\wedge, \vee$ are exchanged and $\top, \bot$ are exchanged and the final states are Q. (That is, the final states are complemented, but since the empty word is not to be accepted, we remove the initial state q_0' from the final states.) It is clear that $\neg A(GQ)$ is a one-clock ATA recognising the complement of the language accepted by $A(GQ)$.

(iii) Next, let D be a formula having syntactic occurrences $GQ_1, \ldots, GQ_k$ of guarded modalities. Using from above the ATAs $A(GQ_i)$ and $\neg A(GQ_i)$, for each i, we construct the one-clock ATA $A(D)$.

Let $GW = \{p_1, \ldots, p_k\}$ be fresh witness propositions. Replace each occurrence of a guarded modality GQ_i by witness formula $\lceil p_i \rceil^0$ in D to get the ITL formula DW (without measurements) such that $D = DW[GQ_i/\lceil p_i \rceil^0, i = 1..k]$.

Let $A(DW) = (Q, \delta, q_0, F)$ be the deterministic finite automaton accepting finite nonempty words over the extended alphabet $2^{Pvar \cup GW}$ which are models of DW. Let $A(GQ_i) = (Q_i, \delta_i, s_i, F_i)$ and $\neg A(GQ_i) = (Q_i', \delta_i', s_i', F_i')$. We construct the one-clock ATA $A(D) = (Q'', \delta'', q_0'', F'')$ as follows. The states Q'' are a disjoint union of the states Q of $A(DW)$ together with the states Q_i and Q_i' of $A(GQ_i)$ and $\neg A(GQ_i)$. The initial state is $q_0'' = q_0$. Final states are also unions of the final states of the component automata $A(DW)$, $A(GQ_i)$ and $\neg A(GQ_i)$.

For each $P \subseteq Pvar$ we have in $A(D)$:

$$\delta''(q, P, true) = \bigvee_{P \subseteq S \subseteq P \cup GW} \left((\delta(q, S), \overline{x}) \wedge \bigwedge_{p_i \in S} \delta_i(s_i, P, true) \wedge \bigwedge_{p_i \notin S} \delta_i'(s_i', P, true) \right)$$

By induction on the number of occurrences of guarded modalities k, we now show that $A(D)$ recognises the θ satisfying D.

For the base case, if there are no guarded modalities and no witnesses, then $A(D) = A(DW)$ does accept models of $DW = D$.

For the induction step, consider $D = DW[GQ/\lceil p \rceil^0]$ for an additional guarded modality GQ and witness proposition p.

Suppose θ is a model of D. Then there is a corresponding model θ_p of DW over the alphabet $2^{Pvar \cup \{p\}}$ which is accepted by $A(DW)$, determining a subset S at every point along the word. The transition function $\delta(q, S)$ determines $\delta''(q, S \cap Pvar)$, which ensures that the corresponding constraint is checked by $A(GQ)$ or $\neg A(GQ)$ and the substituted model θ is accepted by $A(D)$.

Conversely, suppose θ is a word accepted by $A(D)$. At each point of the word, we can ask whether the point formula GQ holds or not. Substituting GQ by p, this defines a set of models of DW over the alphabet $2^{Pvar \cup \{p\}}$. Each such model θ_p is accepted by $A(DW)$, and the $\wedge$-branches in the transition function δ'' ensure, using the automata $A(GQ)$ and $\neg A(GQ)$, that the corresponding timing constraint holds so that θ_p with the valuation of p removed, i.e. θ, is a model of $DW[GQ/\lceil p \rceil^0] = D$.

Acknowledgements. Four FORMATS referees commented on this paper. After reading what they said, we wondered whether we should have called this paper an *overdose*! This is not to criticise the referees: we found their detailed reports constructive and helpful, and would like to thank them. We have removed some of the proofs from this conference version and are preparing a full version.

References

[AD94] R. Alur and D. Dill. A theory of timed automata, *TCS* 126, 1994, 183–236.

[AFH96] R. Alur, T. Feder and T. Henzinger. The benefits of relaxing punctuality, *J.ACM* 43(1), 1996, 116–146.

[Bac03] M. Baclet. Logical characterization of aperiodic data languages, Research Report LSV-03-12 (ENS Cachan, 2003), 16 pp.

[BPT03] P. Bouyer, A. Petit and D. Thérien. An algebraic approach to data languages and timed languages, *Inf. Comput.* 182(2), 2003, 137–162.

[GO] V. Goranko and M. Otto. Model theory of modal logic, in *Handbook of modal logic*, in preparation.

[Gue00] D.P. Guelev. Probabilistic neighbourhood logic, *Proc. FTRTFT*, Pune, *LNCS* 1926 (M. Joseph, ed.), 2000, 264–275.

[HR99] Y. Hirshfeld and A. Rabinovich. A framework for decidable metrical logics, *Proc. ICALP*, Prague, *LNCS* 1644 (J. Wiedermann, P. van Emde Boas and M. Nielsen, eds.), 1999, 422–432.

[IK87] N. Immerman and D. Kozen. Definability with bounded number of bound variables, *Proc. LICS*, Ithaca (IEEE, 1987).

[Imm98] N. Immerman. *Descriptive complexity* (Springer, 1998).

[Kamp68] J.A.W. Kamp. *Tense logic and the theory of linear order*, PhD thesis (UCLA, 1968).

[Koy90] R. Koymans. Specifying real-time properties with metric temporal logic, *Real-time systems* 2(4), 1990, 255–299.

[LW05] S. Lasota and I. Walukiewicz. Alternating timed automata, *Proc. Fossacs*, Edinburgh, *LNCS* 3441 (V. Sassone, ed.), 2005, 250–265.

[LL66] H. Läuchli and J. Leonard. On the elementary theory of linear order, *Fund. Math.* 59, 1966, 109–116.

[Lod00] K. Lodaya. Sharpening the undecidability of interval temporal logic, *Proc. Asian*, Penang, *LNCS* 1961 (J. He and M. Sato, eds.), 2000, 290–298.

[OW05] J. Ouaknine and J. Worrell. On the decidability of metric temporal logic, *Proc. LICS*, Chicago (IEEE, 2005), 188–197.

[Pan96] P.K. Pandya. Weak chop inverses and liveness in mean-value calculus, *Proc. FTRTFT*, Uppsala, *LNCS* 1135 (B. Jonsson and J. Parrow, eds.), 1996, 148–167.

[Pan02] P.K. Pandya. Interval duration logic: expressiveness and decidability, *Proc. TPTS*, Grenoble, *ENTCS* 65(6) (E. Asarin, O. Maler and S. Yovine, eds.), 2002, 19 pp.

[Ven90] Y. Venema. Expressiveness and completeness of an interval tense logic, *Notre Dame J.FL* 31(4), 1990, 529–547.

[Ven91] Y. Venema. A modal logic for chopping intervals, *J. Logic Comput.* 1(4), 1991, 453–476.

[Wil94] T. Wilke. Specifying timed state sequences in powerful decidable logics and timed automata, *Proc. FTRTFT*, Lübeck, *LNCS* 863 (H. Langmaack, W.P. de Roever and J. Vytopil, eds.), 1994, 694–715.

[ZH04] Zhou C. and M.R. Hansen. *Duration calculus* (Springer, 2004).

[ZL94] Zhou C. and Li X. A mean value calculus of durations, in A.W. Roscoe (ed.), *A classical mind: Essays in honour of C.A.R. Hoare* (Prentice-Hall, 1994), 431–451.