Università degli studi di Udine
Corso per il dottorato di ricerca:
Temporal Logics: Satisfiability Checking, Model Checking, and Synthesis
January 2017
Lecture 04: Continuous Temporal Logics

Guest lecturer: Prof. Mark Reynolds

School of Computer Science and Software Engineering
The University of Western Australia

Udine 2017

Dept CS, Maths and Physics,
University of Udine
Friday 20th January 2017
2:30pm until 5:30pm

Outline

1. Introduction
2. RTL
3. Decidability
4. Axioms
5. Mosaics
6. Complexity
7. Tableau
8. Synthesis
9. Model Checking
10. Conclusion and Future Work

What we want and why

Consider the temporal logic of until and since over the real numbers model of time. This logic is an important basis for reasoning about concurrency, metric constraints and planning.

Despite its usefulness and long history, there are no existing implementable reasoning techniques for it.

We look at algorithms for deciding satisfiability, model checking and synthesis.

The Logics: structures and syntax

Fix a countable set $\mathcal{L}$ of propositional atoms.

Definition

Frames $(T, <)$, or *flows of time*, will be irreflexive linear orders. *Structures* $\mathcal{R} = (T, <, h)$ will have a frame $(T, <)$ and a *valuation* h for the atoms i.e. for each atom $p \in \mathcal{L}$, $h(p) \subseteq T$.

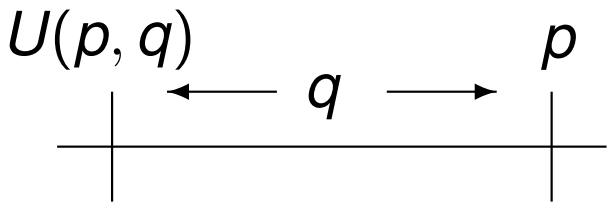
The language $L(U, S)$ is generated by the 2-place connectives U (Until) and S (Since) along with classical $\neg$ and $\wedge$.
Set of formulas contains the atoms and for formulas α and β we include $\neg\alpha$, $\alpha \wedge \beta$, $U(\alpha, \beta)$ and $S(\alpha, \beta)$.
i.e. prefix versions of U and S: $U(p, q)$ says that “until p is true, q holds".

Semantics

Formulas evaluated at points in structures $\mathcal{R} = (T, <, h)$. Usually here $(T, <) = (\mathbb{R}, <)$.

$\mathcal{R}, x \models \alpha$ means α is true at the point $x \in T$:

$\mathcal{R}, x \models p$	iff	$x \in h(P)$, for p atomic;
$\wedge, \neg$		as usual
$\mathcal{R}, x \models U(\alpha, \beta)$	iff	there is $y > x$ in T such that $\mathcal{R}, y \models \alpha$ and for all $z \in T$ such that $x < z < y$ we have $\mathcal{R}, z \models \beta$; and
$\mathcal{R}, x \models S(\alpha, \beta)$	iff	there is $y < x$ in T such that $\mathcal{R}, y \models \alpha$ and for all $z \in T$ such that $y < z < x$ we have $\mathcal{R}, z \models \beta$.



Abbreviations include the usual classical and temporal ones such as $F\alpha \equiv U(\alpha, \top)$, $G\alpha \equiv \neg F \neg \alpha$, $C^+\alpha \equiv U(\top, \alpha)$, $K^+\alpha \equiv \neg C^+(\neg\alpha)$, P, H, C^-, K^-.

Satisfiability of a formula means that there is a model and a time point when that formula is true. *Validity* defined as usual.

When $T = \mathbb{R}$ we call the logic RTL.

So we have the RTL-SAT problem.

Decidability

Eg, decide if

$$U(\top, p) \wedge F\neg p \wedge \neg U(\neg p \vee K^{+}(\neg p), p)$$

is satisfiable.

Decidability of RTL proved by [BG85].

Rabin's decision procedure for the second-order monadic logic of two successors [Rab69] is used in [BG85] to show that that RTL is decidable. One of the two decision procedures in that paper just gives us a non-elementary upper bound on the complexity of RTL-SAT.

Axioms

- all classical tautologies,
- the six Burgess-Xu axioms as follows:

$$G(p \to q) \to (U(p,r) \to U(q,r))$$
$$G(p \to q) \to (U(r,p) \to U(r,q))$$
$$p \wedge U(q,r) \to U(q \wedge S(p,r), r)$$
$$U(p,q) \to U(p, q \wedge U(p,q))$$
$$U(q \wedge U(p,q), q) \to U(p,q)$$
$$U(p,q) \wedge U(r,s) \to$$
$$U(p \wedge r, q \wedge s) \vee U(p \wedge s, q \wedge s) \vee U(q \wedge r, q \wedge s)$$

  along with each of their duals,
- plus axioms for density and no end points: $K^+\top$,$K^-\top$,$F\top$ and $P\top$

and more ...

Axioms Continued

Two for Dedekind completeness:

Prior-U: $U(\top, p) \wedge F\neg p \rightarrow U(\neg p \vee K^{+}(\neg p), p)$

Prior-S: $S(\top, p) \wedge P\neg p \rightarrow S(\neg p \vee K^{-}(\neg p), p)$

and Sep:

Sep: $K^{+}p \wedge \neg K^{+}(p \wedge U(p, \neg p)) \rightarrow K^{+}(K^{+}p \wedge K^{-}p)$

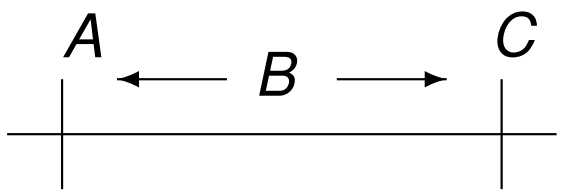
Mosaics

The PSPACE RTL satisfiability decision procedure in [Rey10a] uses linear time mosaic techniques.

Mosaics [Nő5, MMR00, Rey03, HHM+99] are small pieces of a model, in our case, a small piece of a linear-flowed structure.

A mosaic = (subfmlas true at first point, subfmlas true all times in between, subfmlas true at 2nd point)

Syntactically defined with some simple closure properties.



Mosaic Closure Set

Our mosaics will only be concerned with a finite set of formulas:

Definition

For each formula ϕ, define the *closure* of ϕ to be $\mathrm{Cl}\phi = \{\psi, \neg\psi \mid \psi \leq \phi\}$ where $\chi \leq \psi$ means that χ is a subformula of ψ.

We can sometimes think of $\mathrm{Cl}\phi$ as being closed under negation: we could treat $\neg\neg\alpha$ as if it was α.

Mosaic

Definition

Suppose ϕ is from $L(U,S)$. A ϕ-*mosaic* is a triple (A,B,C) of subsets of $\mathrm{Cl}\phi$ such that:

C0.1 A and C are maximally propositionally consistent, and

C0.2 for all $\beta \in \mathrm{Cl}\phi$ with $\neg\beta \in \mathrm{Cl}\phi$ we have $\neg\beta \in B$ iff $\sim\beta \in B$

and the following four *coherency* conditions hold:

C1. if $\neg U(\alpha,\beta) \in A$ and $\beta \in B$ then we have both:

C1.1. $\neg\alpha \in C$ and either $\neg\beta \in C$ or $\neg U(\alpha,\beta) \in C$; and

C1.2. $\neg\alpha \in B$ and $\neg U(\alpha,\beta) \in B$.

C2. if $U(\alpha,\beta) \in A$ and $\neg\alpha \in B$ then we have both:

C2.1 either $\alpha \in C$ or both $\beta \in C$ and $U(\alpha,\beta) \in C$; and

C2.2. $\beta \in B$ and $U(\alpha,\beta) \in B$.

C3-4 mirror images of C1-C2.

An example mosaic

An example mosaic for $\phi = Gp \wedge C^+(K^+q \wedge K^+(\neg q))$ is

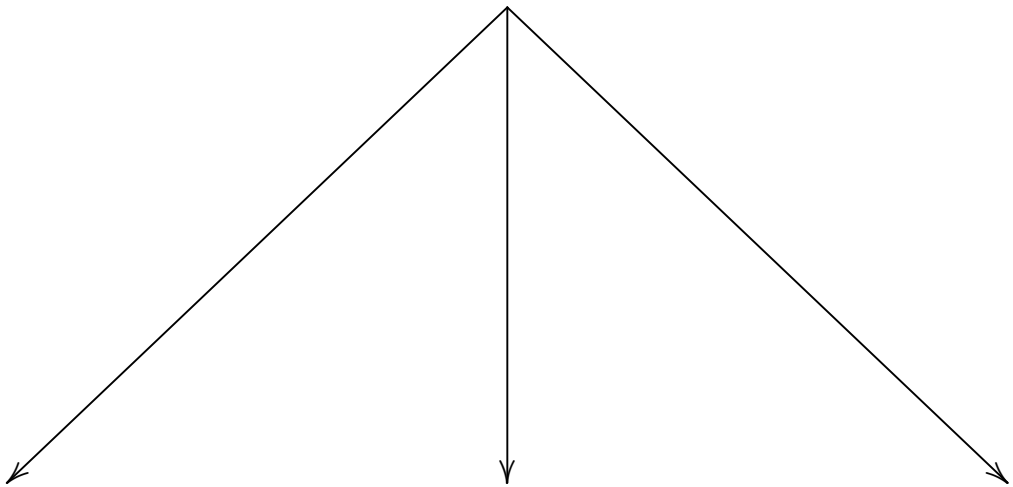
$$(\{p, q, Gp, ..., \phi\},$$
$$\{p, Gp, K^+q, K^+(\neg q), ...\},$$
$$\{p, q, Gp, ..., \neg\phi\}).$$

Coherency conditions apply. Eg, Gp in the start implies p, Gp in the cover and p, Gp in the end.

Mosaics can be decomposed

The example mosaic for $\phi = Gp \wedge C^+(K^+q \wedge K^+(\neg q))$ decomposes

$(\{p, q, Gp, ..., \phi\},$
$\{p, K^+q, K^+(\neg q), ...\},$
$\{p, q, Gp, ..., \neg\phi\}).$



$(\{p, q, Gp, ..., \phi\},$
$\{p, K^+q, K^+(\neg q), ...\},$
$\{p, q, Gp, K^+q, ..., \}).$

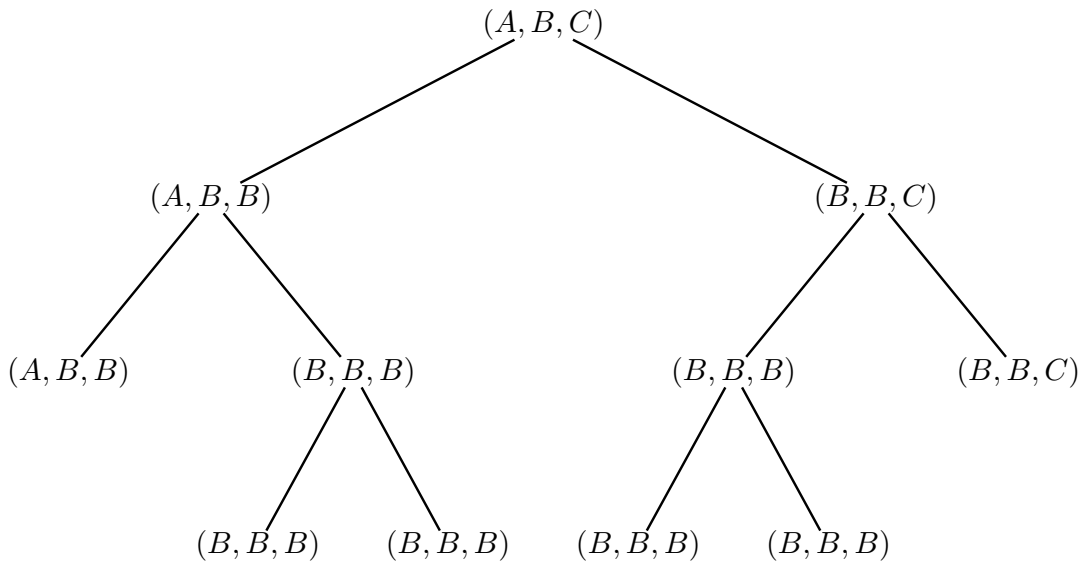
$(\{p, q, Gp, ..., \phi\},$
$\{p, K^+q, K^+(\neg q), ...\},$
$\{p, q, Gp, ..., \neg\phi\}).$

$(\{p, q, Gp, ..., \phi\},$
$\{p, K^+q, K^+(\neg q), ...\},$
$\{p, q, Gp, ..., \neg\phi\}).$

Full decomposition means defects witnessed. Eg $\neg q$ not in cover.

Tableaux or tree like structure

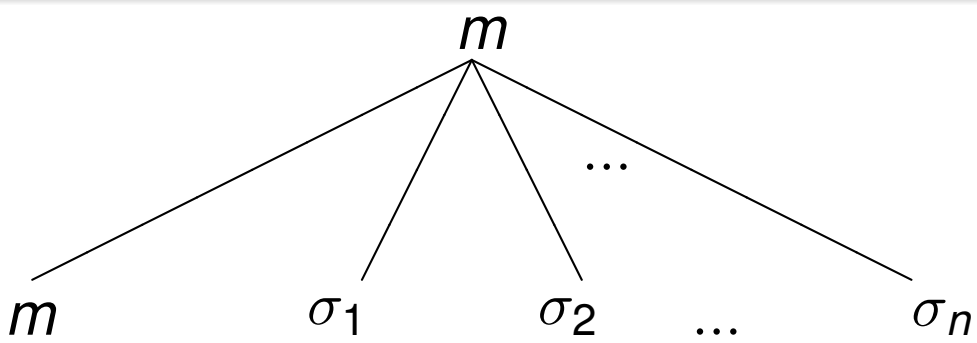
We keep going to make a tree-shaped tableau of mosaics.



Patterns appear

Definition

We say that m is fully decomposed by the *tactic* $\text{lead}\ \sigma$, for some sequence σ of mosaics iff $\langle m \rangle \,^\frown \sigma$ is a full decomposition of m.



The $\text{trail}\ \sigma$ tactic is mirror.

Also a repetitive pattern called a *shuffle*.

Look at tree of patterns instead ...

Sketch RTLSAT Proof From [Rey10a]

Guess (A, B, C) for ϕ and then check that (A, B, C) is satisfiable.

In [Rey10a] satisfiability is checked via a tree of decompositions via tactics (covers getting fuller as you go down branches).

Immediately gives an EXPTIME decision procedure a la [Pra79].

By being clever in choosing decompositions, bounding the depth of a tree, can get a PSPACE result.

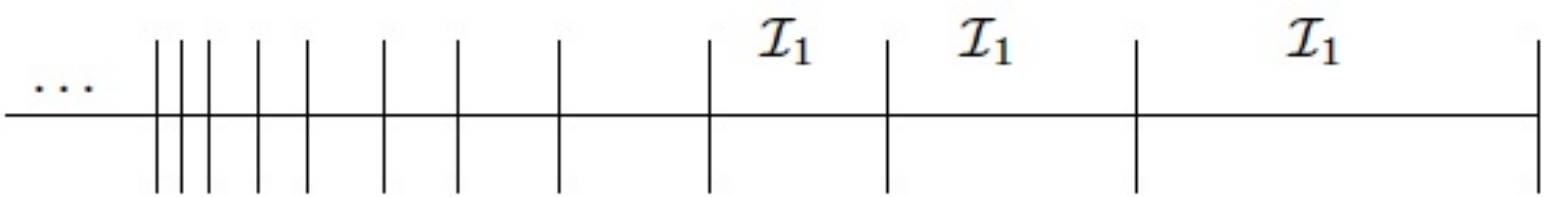
Theorem

RTLSAT is in PSPACE.

But the resulting tree (in case of a positive answer) looks like a sort of description of a model ...

Lead

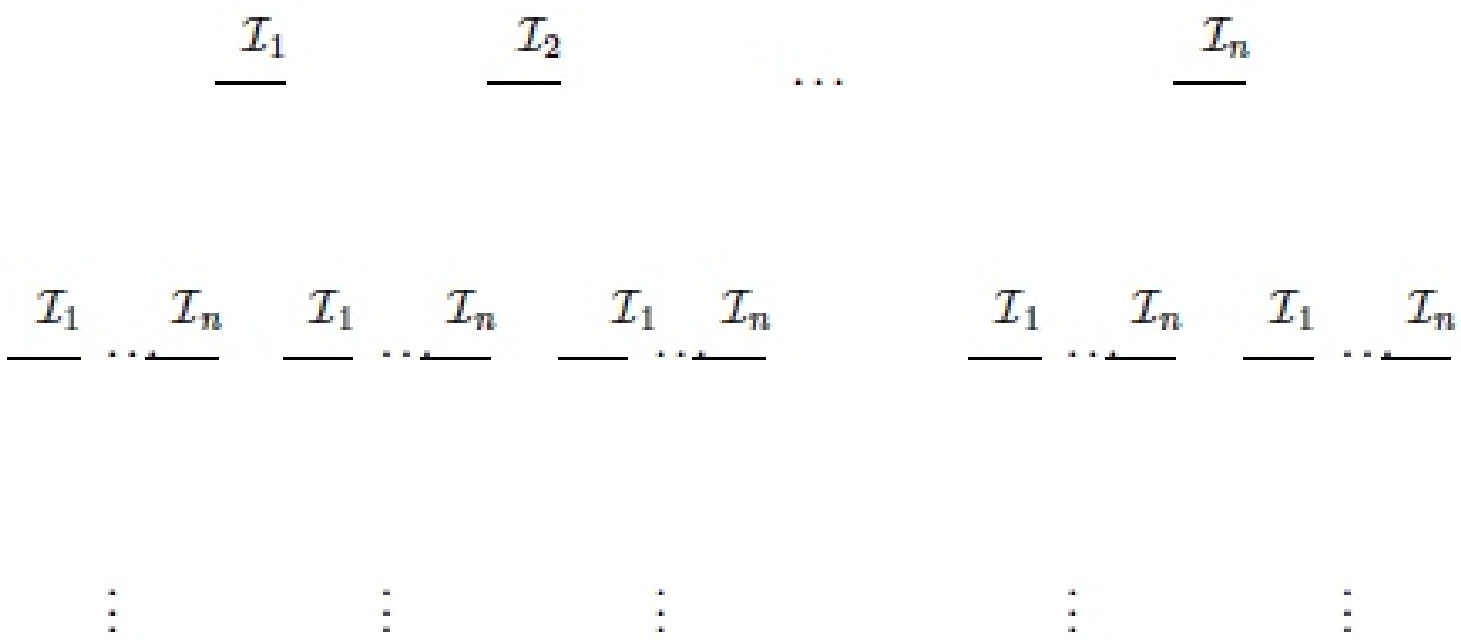
The *lead* operation, $\mathcal{I} = \overleftarrow{\mathcal{I}_1}$ corresponds to ω submodels, each corresponding to $\mathcal{I}$, and each preceding the last, ...



The *trail* operator is the mirror image of the *lead* operation, whereby $\mathcal{I} = \overrightarrow{\mathcal{I}_1}$ corresponds to ω structures, each corresponding to $\mathcal{I}_1$ and each proceeding the earlier structures.

Shuffle

The *shuffle* operator is harder to represent with a diagram. The model expression $\mathcal{I} = \langle \mathcal{I}_1, \ldots \mathcal{I}_n \rangle$ corresponds to a dense, thorough mixture of intervals corresponding to $\mathcal{I}_1, \ldots, \mathcal{I}_n$, without endpoints.

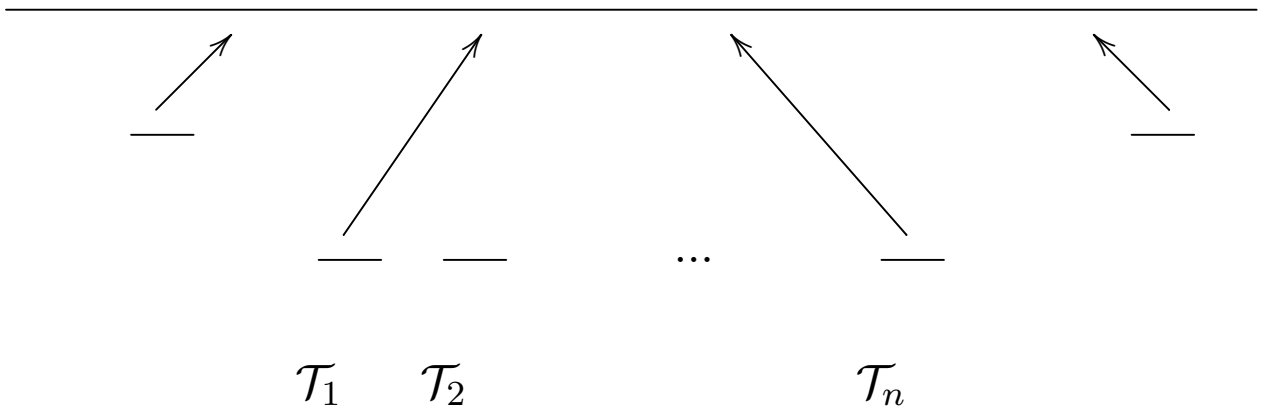


The shuffle operation, where $\mathcal{I} = \langle \mathcal{I}_1, \ldots, \mathcal{I}_n \rangle$

Shuffles

Definition

Suppose that $(T_1, <_1, g_1), ..., (T_n, <_n, g_n)$ are linear structures. Further, suppose there is a non-empty, non-singleton linear order $(B, <_B)$ and a map $\pi : B \to \{1, ..., n\}$ such that for all $b < c$ from B, for all $j \in \{1, ..., n\}$, there is $d \in B$ with $b < d < c$ and $\pi(d) = j$. Then $\mathcal{T} = \Sigma_{b \in (B,<)} \pi(b)$ is a *shuffle* of $\{(T_i, <_i, g_i) | i = 1, ..., n\}$.



Complexity

[Rey10a] proved that RTL is PSPACE-complete (so no more complex than $L(U, S)$ over $\mathbb{N}$ [SC85]).

Also [Rey10b] proved US over all linear orders is PSPACE-complete.

Tableau

The tableau-like structure we saw as part of the complexity proof is the basis for a tableau for RTL but we need further work.

Define weight of a ϕ-mosaic m as $|\mathrm{Cl}(\phi) \setminus \mathrm{cover}(m)|$. So, if $|\mathrm{Cl}(\phi)| = L$ then for every ϕ-mosaic m, $0 \leq \mathrm{weight}(m) \leq L$, although, as the cover of a mosaic over the reals can not be inconsistent, the weight must be at least about half of L. Note also that L is bounded by the twice the length of ϕ.

Note that weight does not increase as you travel along a branch from the root: the cover of child labels is a superset of the cover of their parent.

We need to check properties of the set of nodes labelled with mosaics of a particular weight. Suppose we are looking at weight $k > 0$.

Look for a part of the tableau with these properties

Definition

Mosaic tableau T is a $\mathbb{R}$-tableau iff it satisfies the following conditions:

1. T has no units, i.e. every mosaic has a decomposition into at least two mosaics;
2. T has no central sticks;
3. T has no shuffles without a singleton; and
4. T only has shuffles with concise edges.

These are essentially simple graph-theoretic properties of the labels on the decomposition tree but we will not define them in this talk.

Tableau Results

Theorem

$L(U, S)$ formula ϕ is $\mathbb{R}$-satisfiable iff ϕ has a successful $\mathbb{R}$-tableau.

By guessing a tableau of double exponential size we have a decision procedure that runs in 2-NEXPTIME.

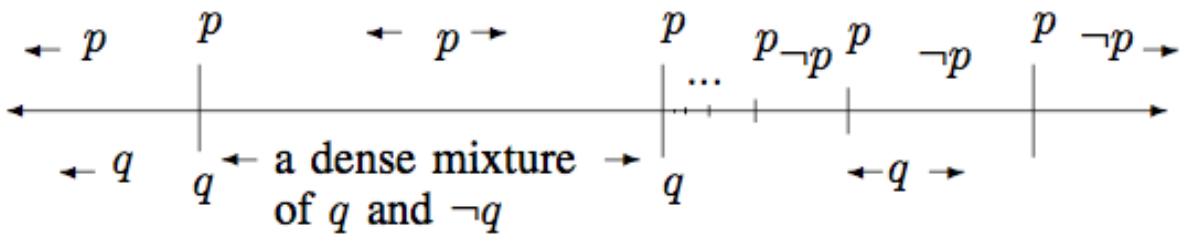
Synthesis: What we want

We want an algorithm which does the following ...

Given a satisfiable formula such as
$\phi = U(q \wedge K^{+}p \wedge GS(p, \neg p), \neg U(q, \neg q) \wedge \neg U(q, q))$.

Output a finite description of a model (any model) of ϕ.

Eg picture ...



Model Expressions

First, though, any synthesis algorithm will need to output the description of a particular temporal structure over the reals (although we don't care up to order-preserving isomorphism). Model Expressions are an abstract syntax for defining (general linear) models that are constructed using the following set of primitive operators:

$$\mathcal{I} ::= a \mid \mathcal{I} + \mathcal{J} \mid \overrightarrow{\mathcal{I}} \mid \overleftarrow{\mathcal{I}} \mid \langle \mathcal{I}_0, ..., \mathcal{I}_n \rangle$$

where $a \in \Sigma = \wp(\mathcal{L})$
so letter indicates the atoms true at a point.
We refer to these operators, respectively, as a letter, concatenation, lead, trail, and shuffle.

Correspondence

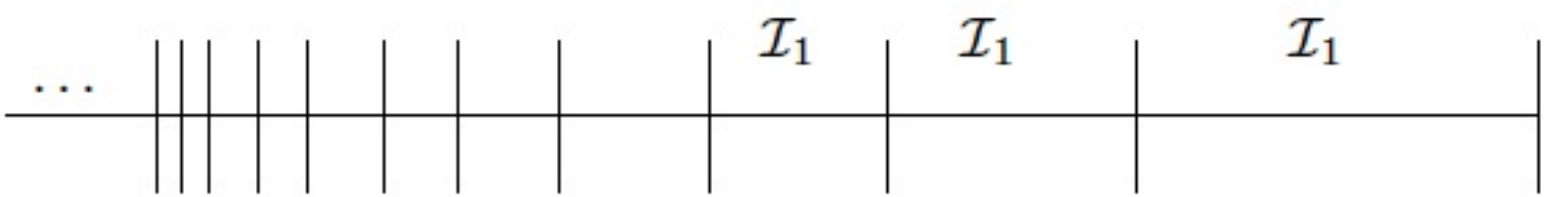
Definition

A model expression $\mathcal{I}$ *corresponds* to a structure as follows:

- λ is the empty expression and corresponds to the empty structure $(\emptyset, <, h)$ where $<$ is the empty relation and $h(p) = \emptyset$ for all $p \in \mathcal{L}$.
- a corresp. to a single point structure $(\{x\}, <, h)$ where x is any object, $<$ is the empty relation and $h(p) = \{x\}$ if and only if $p \in a$.
- $\mathcal{I} + \mathcal{J}$ corresponds to any structure isomorphic to $\mathbf{T} \oplus \mathbf{S}$, for some structure $\mathbf{T}$ which corresponds to $\mathcal{I}$ and $\mathbf{S}$ which corresponds to $\mathcal{J}$.
- $\overleftarrow{\mathcal{I}}$ corresp. to any structure iso. to $\Sigma_{(\mathbb{N},>)}\mathcal{X}_t$ where, for all $t \in \mathbb{N}$, $\mathcal{X}_t = \mathcal{X}$ is some structure corresponding to $\mathcal{I}$. (Trail is mirror).
- For the case of shuffle, say $\mathcal{I} = \langle \mathcal{I}_1, ..., \mathcal{I}_n \rangle$, and suppose that for each $i = 1, ..., n$, $\mathcal{X}_i$ corresponds to $\mathcal{I}$. Now define $s : \mathbb{Q} \to \langle \mathcal{I}_1, ..., \mathcal{I}_n \rangle$ by: if $t \in Q_i \subseteq \mathbb{Q}$ then $s(t) = \mathcal{X}_i$; otherwise—if $t \in \mathbb{Q} \setminus \bigcup_{i \leq n} Q_i$—define $s(t) = \mathcal{X}_1$. Then $\mathcal{I}$ corresponds to any structure isomorphic to $\Sigma_{(\mathbb{Q},<)} s(t)$.

Lead

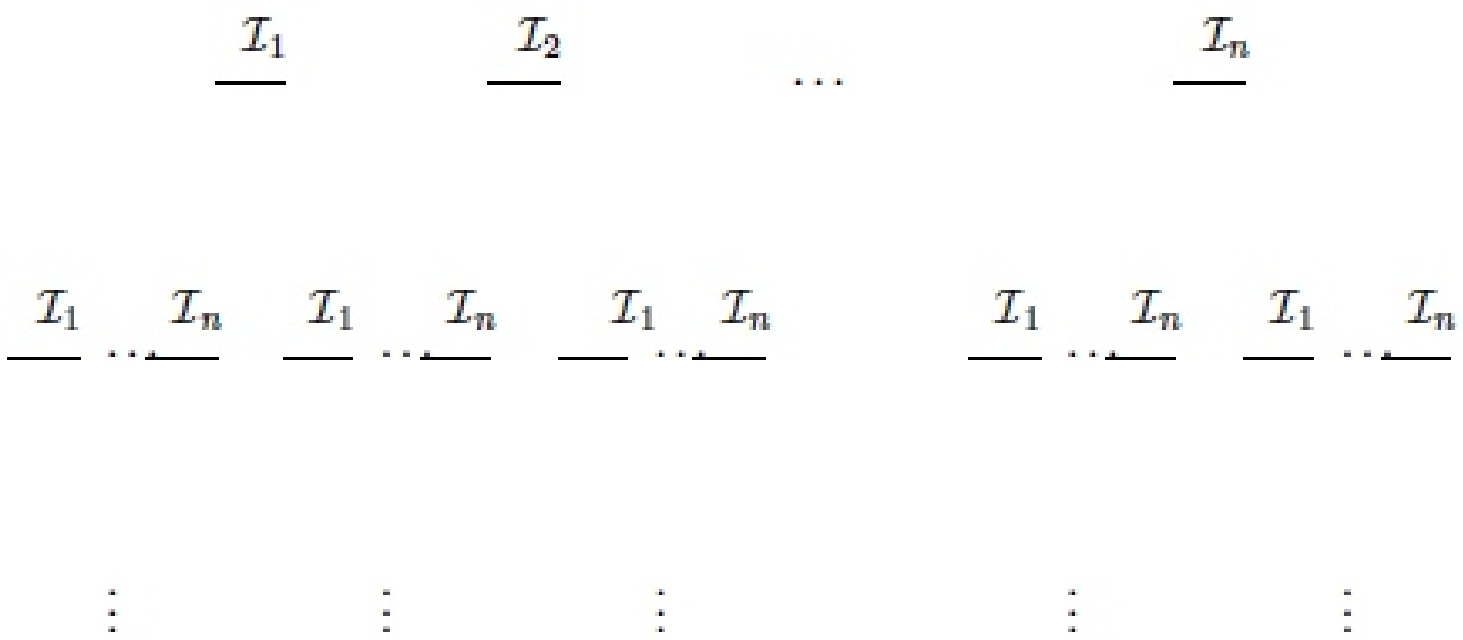
The *lead* operation, $\mathcal{I} = \overleftarrow{\mathcal{I}_1}$ corresponds to ω submodels, each corresponding to $\mathcal{I}$, and each preceding the last, ...



The *trail* operator is the mirror image of the *lead* operation, whereby $\mathcal{I} = \overrightarrow{\mathcal{I}_1}$ corresponds to ω structures, each corresponding to $\mathcal{I}_1$ and each proceeding the earlier structures.

Shuffle

The *shuffle* operator is harder to represent with a diagram. The model expression $\mathcal{I} = \langle \mathcal{I}_1, \ldots \mathcal{I}_n \rangle$ corresponds to a dense, thorough mixture of intervals corresponding to $\mathcal{I}_1, \ldots, \mathcal{I}_n$, without endpoints.



The shuffle operation, where $\mathcal{I} = \langle \mathcal{I}_1, \ldots, \mathcal{I}_n \rangle$

RMEs

Only want to build separable, dense struct. w/out endpoints...

Definition (Real Model Expression)

$$\mathcal{K} ::= \langle a_0, ..., a_m, x_1 + \mathcal{K}_1 + y_1, ..., x_n + \mathcal{K}_n + y_n \rangle \mid \mathcal{K}_0 + a + \mathcal{K}_1 \mid \overrightarrow{a + \mathcal{K}} \mid \overleftarrow{\mathcal{K} + a}$$

where $a, a_i, x_i, y_i \in \wp\mathcal{L}$, and $m, n \geq 0$

The letter a_0 can be later used as a sort of background filler to ensure that the shuffle is Dedekind complete.

RMEs always define open intervals. Base element of recursion is shuffle with only points. Will define a dense, separable linear order with all letters homogeneously distributed.

Non correspondence

Lemma

Every real model expression corresponds to some structure whose frame is dense, separable and without end-points.

However, ...
These structures do not have a real frame: they're countable. Luckily, there is an iterative way of constructing a particular real-flowed structure for each RME...

$R(\mathcal{I})$

Definition ($\mathcal{R}(\mathcal{I})$)

Suppose $\mathcal{I}$ is a model expression. We define a particular structure $\mathcal{R}(\mathcal{I})$ inductively and depending on the form of $\mathcal{I}$ as follows.

- $\mathcal{R}(\lambda) = (\emptyset, \emptyset, h)$, where $h(p) = \emptyset$ for every $p \in \mathcal{L}$.
- For a letter a, $\mathcal{R}(a) = (\{0\}, \emptyset, h)$, where for each $p \in \mathcal{L}$, $h(p) = \{0\}$ if $p \in a$ and $h(p) = \emptyset$ otherwise.
- If $\mathcal{I}_1$ and $\mathcal{I}_2$ are model expressions, then $\mathcal{R}(\mathcal{I}_1 + \mathcal{I}_2) = \mathcal{R}(\mathcal{I}_1) \oplus \mathcal{R}(\mathcal{I}_2)$.
- If $\mathcal{I}$ is a model expression then $\mathcal{R}(\overleftarrow{\mathcal{I}}) = \Sigma_{(\mathbb{N},>)}\mathcal{X}_t$ where $\mathcal{X}_t = \mathcal{R}(\mathcal{I})$ for each $t \in \mathbb{N}$.
- $\mathcal{R}(\overrightarrow{\mathcal{I}})$ is analogously based on $(\mathbb{N}, <)$.
- For the case of shuffle, say $\mathcal{I} = \langle \mathcal{I}_1, ..., \mathcal{I}_n \rangle$, define $f : \mathbb{R} \to \langle \mathcal{I}_1, ..., \mathcal{I}_n \rangle$ by: if $t \in Q_i \subseteq \mathbb{Q}$ then $f(t) = \mathcal{I}_i$; otherwise—if $t \in \mathbb{R} \setminus \bigcup_{i \leq n} Q_i$—define $f(t) = \mathcal{I}_1$. Define $\mathcal{R}(\mathcal{I}) = \Sigma_{(\mathbb{R},<)}\mathcal{R}(f(t))$.

Actually a real model

Lemma

For every real model expression $\mathcal{K}$, $R(\mathcal{K})$ is a structure with a frame that is isomorphic to the reals.

Synthesis for RTL

Results from French, McCabe-Dansted and Reynolds [FMDR13].

Theorem

There is an FPSPACE procedure which given a formula ϕ from $L(U, S)$ will decide whether ϕ is R-satisfiable or not and, if so, will provide a real model expression for a model of ϕ.

Theorem

The complexity of the problem of providing a real model expression for a model of formulas of L(U,S) is FPSPACE-complete.

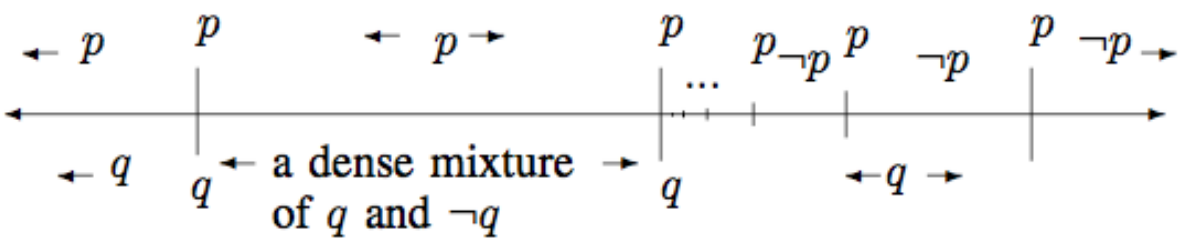
Example

We now have an algorithm which does the following ...

Given a satisfiable formula such as
$\phi = U(q \wedge K^{+}p \wedge GS(p, \neg p), \neg U(q, \neg q) \wedge \neg U(q, q))$.

It outputs finite description of some model of ϕ.

$$\langle \{p, q\} \rangle + \{p, q\} + \langle \{p, q\}, \{p\} \rangle + \{p, q\} + \overleftarrow{\{p, q\} + \langle \{q\} \rangle}$$



Regular Model Expressions

Regular Model Expressions (RegMEs) are similar to model expressions but also allow the *Kleene star* "*" and *or* "|" operators, resulting in the following syntax:

$$\mathcal{I} ::= \nu \mid \lambda \mid (\mathcal{I} + \mathcal{I}) \mid (\mathcal{I}|\mathcal{I}) \mid \mathcal{I}* \mid \overleftarrow{\mathcal{I}} \mid \overrightarrow{\mathcal{I}} \mid \langle \mathcal{I}_1, \ldots, \mathcal{I}_n \rangle$$

RegME Accepting a structure

MEs *correspond* to a particular structures that are all elementary equivalent. However, RegMEs are more like automata in that the structures accepted by a single RegMEs need not be elementary equivalent. For this reason we use the term “accept” rather than “correspond”, unlike previous papers that only deal with MEs.

RegME Accepting a structure (continued)

Definition

A model expression $\mathcal{I}$ *accepts* structures as follows:

- λ is the empty expression and accepts the empty structure $(\emptyset, <, h)$ where $<$ is the empty relation and $h(p) = \emptyset$ for all $p \in \mathcal{L}$.
- ν accepts any single point structure $(\{x\}, <, h)$ where x is any object, $<$ is the empty relation and $h(p) = \{x\}$ if and only if $p \in \nu$.
- $\mathcal{I} + \mathcal{J}$ accepts any structure isomorphic to $\mathbf{T} \oplus \mathbf{S}$, for some structure $\mathbf{T}$ which is accepted by $\mathcal{I}$ and $\mathbf{S}$ which is accepted by $\mathcal{J}$.
- $\overleftarrow{\mathcal{I}}$ accepts any structure isomorphic to $\Sigma_{(\mathbb{N},>)} X_t$ where, for all $t \in \mathbb{N}$, $X_t = X$ is some structure accepted by $\mathcal{I}$. (trail mirror)
- $\langle \mathcal{I}_1, \ldots, \mathcal{I}_n \rangle$ accepts a structure if it is isomorphic to some $\Sigma_{(T,<)} X_t$ where: $(T, <)$ is a dense non-empty linear order without end-points; there is a function f from T to $\{1, \ldots, n\}$ such that for every x, $y \in T$, for every $i \in \{1, \ldots, n\}$, if $x < y$ then there is $z \in T$ such that $x < z < y$ and $f(z) = i$; additionally for all $t \in T$ we have X_t accepted by $\mathcal{I}_{f(t)}$.

RegME Accepting a structure (continued)

Definition

Now we define the regular operators:

- $\mathcal{I}|\mathcal{J}$ accepts any structure that is accepted by either $\mathcal{I}$ or $\mathcal{J}$.
- $\mathcal{I}*$ accepts any structure that accepted by λ, $\mathcal{I}$, $\mathcal{I}+\mathcal{I}$, or $\mathcal{I}+\mathcal{I}+\mathcal{I}$ and so on.

Model Checking

(Regular) Model expressions also allow us to do model checking with RTL.

Definition

We define the *RegME-checking problem* as follows: given a RegME $\mathcal{I}$ and formula ϕ, determine whether $\mathcal{I}$ accepts a structure $\mathbf{T} = (\mathbf{T}, <, \mathbf{h})$ containing an $x \in T$ such that $\mathbf{T}, \mathbf{x} \models \phi$.

Note that when $\mathcal{I}$ accepts all structures, this becomes equivalent to testing satisfiability of ϕ. For example if $\mathcal{I} = \langle \lambda, \emptyset, \{p\}, \{q\}, \{p, q\} \rangle$ then model checking a formula ϕ over the atoms $\{p, q\}$ against $\mathcal{I}$ is equivalent to checking the satisfiability of ϕ. Since satisfiability checking RTL is PSPACE-complete, this may suggest another way of showing that our model checking problem is PSPACE-hard; however, note that the length of the RegME that accepts all structures is exponential in the number of atoms.

Model checking algorithm

From McCabe-Dansted, Reynolds and French, TIME 2016, algorithm reduces the model checking problem into an RTL satisfiability problem.

The basic idea is that satisfiability checking linear time temporal logics is in PSPACE. Thus, we can model check M against ϕ by converting the model M into a formula Ψ_M, and then testing the satisfiability of the conjunction of ϕ and Ψ_M. We will introduce some dummy points, so we will have to modify ψ slightly to skip over those points.

Theorem

Our model checking procedure is sound and complete.

Lemma

Model Checking RegMEs is in PSPACE.

Conclusion and Future Work

We considered synthesizing, or constructing, a monadic structure over the reals, from a given first-order specification.

Presented notation for giving a manageable description of the compositional construction of such a model.

Used mosaics and separation techniques to give an algorithm for synthesis.

Future Work:
Implementation via tableaux (at least the temporal synthesis part).
How many “different" models? Using regular expressions to describe all models.
Adding metric restrictions.

Thank you for listening

Questions? Comments.


J. P. Burgess and Y. Gurevich.
The decision problem for linear temporal logic.
Notre Dame J. Formal Logic, 26(2):115–128, 1985.


R. Hirsch, I. Hodkinson, M. Marx, S. Mikulás, and M. Reynolds.
Mosaics and step-by-step. Remarks on “A modal logic of relations”.
In E. Orlowska, editor, *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa*, volume 24 of *Studies in Fuzziness and Soft Computing*, pages 158–167. Springer-Verlag, 1999.


M. Marx, S. Mikulas, and M. Reynolds.
The mosaic method for temporal logics.
In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, Proceedings of International Conference, TABLEAUX 2000, Saint Andrews, Scotland, July 2000*, LNAI 1847, pages 324–340. Springer, 2000.


I. Németi.

Decidable versions of first order logic and cylindric-relativized set algebras.
In L. Csirmaz, D. Gabbay, and M. de Rijke, editors, *Logic Colloquium '92*, pages 171–241. CSLI Publications, 1995.


V. R. Pratt.
Models of program logics.
In *Proc. 20th IEEE. Symposium on Foundations of Computer Science, San Juan*, pages 115–122, 1979.


M. O. Rabin.
Decidability of second order theories and automata on infinite trees.
American Mathematical Society Transactions, 141:1–35, 1969.


Mark Reynolds.
The complexity of the temporal logic with "until" over general linear time.
J. Comput. Syst. Sci., 66(2):393–426, 2003.


M. Reynolds.

The complexity of the temporal logic over the reals.
Annals of Pure and Applied Logic, 161(8):1063–1096, 2010.
Online at doi:10.1016/j.apal.2010.01.002.


Mark Reynolds.
The complexity of temporal logics over linear time.
Journal of Studies in Logic, 3:19–50, 2010.


A. Sistla and E. Clarke.
Complexity of propositional linear temporal logics.
J. ACM, 32:733–749, 1985.


Rajeev Alur and Thomas A. Henzinger.
Real-time logics: Complexity and expressiveness.
Inf. Comput., 104(1):35–77, 1993.


Burgess, J. P. Axioms for Tense Logic I:“Since” and “Until”, Notre Dame J. Formal Logic, 23(2):367–374, 1982.


J. P. Burgess and Y. Gurevich.
The decision problem for linear temporal logic.
Notre Dame J. Formal Logic, 26(2):115–128, 1985.


Zhou Chaochen, C. A. R. Hoare, and Anders P. Ravn.
A calculus of durations.
Inf. Process. Lett., 40(5):269–276, 1991.


A. Ehrenfeucht.
An application of games to the completeness problem for formalized theories.
Fund. Math., 49:128–141, 1961.


M. Fitting.
Proof methods for modal and intuitionistic logics.
Reidel, 1983.


O. Friedmann and M. Lange.
A solver for modal fixpoint logics.
Electr. Notes Theor. Comput. Sci., ?(?):99–111, 2010.


Roland Frasse.
Sur quelques classifications des systemes de relations, thesis, paris, 1953.

Publications Scientifiques de l'Universite d'Alger, series A 1:35–182, 1954.


French, T., J. McCabe-Dansted and M. Reynolds, *Synthesis and model checking for continuous time: Long version*, Technical report, CSSE, UWA (April 2012), "http://www.csse.uwa.edu.au/∼mark/research/Online/sctm.htm".


Gabbay, D. M. and I. M. Hodkinson, *An axiomatisation of the temporal logic with until and since over the real numbers*, J. Logic and Computation **1** (1990), pp. 229 – 260.


D. M. Gabbay, I. M. Hodkinson, and M. A. Reynolds.
Temporal expressive completeness in the presence of gaps.
In J. Oikkonen and J. Väänänen, editors, *Logic Colloquium '90, Proceedings ASL European Meeting 1990, Helsinki*, number 2 in Lecture Notes in Logic, pages 89–121. Springer-Verlag, 1993.


D. Gabbay, I. Hodkinson, and M. Reynolds.
Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 1.

Oxford University Press, 1994.


R. Goré.
Tableau methods for modal and temporal logics.
In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.


Y. Gurevich.
Elementary properties of ordered abelian groups.
Algebra and Logic, 3:5–39, 1964.
(Russian; an English version is in Trans. Amer. Math. Soc. 46 (1965), 165–192).


G. Hughes and M. Cresswell.
An Introduction to Modal Logic.
Methuen, London, 1968.


R. Hirsch and I. Hodkinson.
Relation algebras by games, volume 147 of *Studies in Logic and the Foundations of Mathematics*.

North-Holland, Amsterdam, 2002.


R. Hirsch, I. Hodkinson, M. Marx, S. Mikulás, and M. Reynolds.
Mosaics and step-by-step. Remarks on “A modal logic of relations”.
In E. Orlowska, editor, *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa*, volume 24 of *Studies in Fuzziness and Soft Computing*, pages 158–167. Springer-Verlag, 1999.


J. Halpern and Y. Shoham.
A propositional modal logic of time intervals.
In *Proceedings, Symposium on Logic in Computer Science*. IEEE, Boston, 1986.


H. Kamp.
Tense logic and the theory of linear order.
PhD thesis, University of California, Los Angeles, 1968.


Y. Kesten, Z. Manna, and A. Pnueli.
Temporal verification of simulation and refinement.

In *A decade of concurrency: reflections and perspectives: REX school/symposium, Noordwijkerhout, the Netherlands, June 1–4, 1993*, pages 273–346. Springer–Verlag, 1994.


Martin Lange and Colin Stirling.
Model checking games for branching time logics.
J. Log. Comput., 12(4):623–639, 2002.


Läuchli, H. and J. Leonard, *On the elementary theory of linear order*, Fundamenta Mathematicae **59** (1966), pp. 109–116.


M. Marx, S. Mikulas, and M. Reynolds.
The mosaic method for temporal logics.
In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, Proceedings of International Conference, TABLEAUX 2000, Saint Andrews, Scotland, July 2000*, LNAI 1847, pages 324–340. Springer, 2000.


I. Németi.
Decidable versions of first order logic and cylindric-relativized set algebras.

In L. Csirmaz, D. Gabbay, and M. de Rijke, editors, *Logic Colloquium '92*, pages 171–241. CSLI Publications, 1995.


A. Pnueli.
The temporal logic of programs.
In *Proceedings of the Eighteenth Symposium on Foundations of Computer Science*, pages 46–57, 1977.
Providence, RI.


V. R. Pratt.
Models of program logics.
In *Proc. 20th IEEE. Symposium on Foundations of Computer Science, San Juan*, pages 115–122, 1979.


A. Rabinovich.
On the decidability of continuous time specification formalisms.
Journal of Logic and Computation, 8:669–678, 1998.


Reynolds, M., *An axiomatization for Until and Since over the reals without the IRR rule*, Studia Logica **51** (1992), pp. 165–193.


M. Reynolds and C. Dixon.
Theorem-proving for discrete temporal logic.
In M. Fisher, D. Gabbay, and L. Vila, editors, *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 279–314. Elsevier, 2005.


Mark Reynolds.
The complexity of the temporal logic with "until" over general linear time.
J. Comput. Syst. Sci., 66(2):393–426, 2003.


Mark Reynolds.
Dense time reasoning via mosaics.
In *TIME '09: Proceedings of the 2009 16th International Symposium on Temporal Representation and Reasoning*, pages 3–10, Washington, DC, USA, 2009. IEEE Computer Society.


M. Reynolds.
The complexity of the temporal logic over the reals.
Annals of Pure and Applied Logic, In press, 2010.

Accepted 2009 to appear. Journal version online February 2010 at doi:10.1016/j.apal.2010.01.002.


Mark Reynolds.
The complexity of decision problems for linear temporal logics.
Journal of Studies in Logic, 3:19–50, 2010.


M. Reynolds.
A tableau for until and since over linear time.
Technical report, CSSE, UWA, April 2011.
http://www.csse.uwa.edu.au/ mark/research/Online/uslintab.htm.


Mark Reynolds.
Metric temporal logic revisited.
Technical report, CSSE, UWA, April 2011.
to be submitted to a conference or journal April 2011, online version at
http://www.csse.uwa.edu.au/ mark/research/Online/MRTL.htm.


Rosenstein, J. G., Linear Orderings, Academic Press, New York, 1982.


A. Sistla and E. Clarke.
Complexity of propositional linear temporal logics.
J. ACM, 32:733–749, 1985.