2013 20th International Symposium on Temporal Representation and Reasoning

An algebraic system of temporal structures

Tim French John M^{c}Cabe-Dansted Mark Reynolds
School of Computer Science and Software Engineering, The University of Western Australia
35 Stirling Highway, Crawley WA 6009
Perth, Australia.

***Abstract*—Lauchli and Leonard, in 1966, described a series of operations which are able to build all linear temporal structures up to first order equivalence. More recently these operations have been used to describe executions of continuous systems for the purposes of model checking real-time specifications. In this paper we present an algebra over these operations and show that it is both sound and complete, in that it can generate all equivalences over these models.**

I. Introduction

Standard temporal logics are based on a discrete, natural numbers model of time [1]. However, a dense, continuous or specifically real-numbers model of time may be better for many applications, ranging from philosophical, natural language and AI modelling of human reasoning to computing and engineering applications of concurrency, refinement, open systems, analogue devices and metric information.

One of the main results in the paper [2] was that a formula of *Real-Time Logic* (RTL) has a real-flowed model if has a constructable real model. Here a real-flowed model means a model of linear temporal logic, whose underlying flow of time is represented by the real numbers, and a constructable real model refers to a real model definable using a formal model-building language based on the operations of Lauchli and Leonard [3] (points, concatenation, leads, trails and shuffles, also described in [4]).

In this paper we consider the generalization to arbitrary linear temporal structures. The operations of Lauchli and Leonard are able to generate expressions that correspond to temporal structures in such a way that any satisfiable formula in the monadic second order theory of $<$ is satisfied by such a temporal structure. We refer to the generated expressions as *model expressions*.

We will describe a number of known results for these model expressions: most importantly,

1) every satisfiable temporal logic formula is satisfied by a temporal structure that corresponds to some model expression; and
2) the class of temporal structures that correspond to a given model expression is an isomorphism class.

That is, models expressions are able to finitely describe a satisfying structure for every satisfiable temporal logic formula. We will make this claim precise later. The main result of this paper is to present an algebra over these operations that is adequate for determining whether two model expressions are equivalent. Model expressions are a natural artifact for model-checking and synthesis problems over general linear time, and recent applications of timed and hybrid automata [5] have increased the significance of these problems. It is useful to have a complete set of algebraic identities for model expressions over general linear structures, but perhaps more useful to have a set of such identities over dense continuous structures, over the reals. However, we see in [2] that these results come for free, as we can find a syntactic restriction over the set of all model expressions that guarantees the expression corresponds to the reals. We aim to show that many other common linear orders can also be identified with syntactic restrictions of model expressions.

We anticipate that an algebraic approach for model expressions will be useful in the analysis of continuous systems. Model expressions describing a trace of a continuous system may be simplified, reduced to a canonical form, or proven equivalent to another expression using this algebra.

II. Linear temporal Structures

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

An isomorphism is a bijective mapping from one structure to another that preserves the temporal relation $<$ and the valuation h. This is an important notion of equivalence for us, as we will show that equivalent structures satisfy the same set of formulas in $L(U, S)$.

DEFINITION 1: We say two structures $\mathcal{T} = (T, <, h)$ and $\mathcal{T}' = (T', <', h')$ are isomorphic (written $\mathcal{T} \cong \mathcal{T}'$) if and only if there is a bijection $f\colon T \longrightarrow T'$ where for all $x, y \in T$ $x < y$ if and only if $f(x) <' f(y)$, and for all $p \in P$ $x \in h(p)$ if and only if $f(x) \in h'(p)$. We say $(T, <, h)$ and $(T', <', h')$ are disjoint isomorphic if they are isomorphic and T and T' are disjoint.

To simplify notation below, given a structure $(T, <, h)$, and some $T' \subset T$, we let $(T', <, h)$ refer to the structure $(T', <^{T'}, h^{T'})$ where $<^{T'}$ is the restriction of $<$ to the set T' and $h^{T'}$ is the restriction of h to the domain T'. Isomorphisms between structures preserve formulas of temporal logic just as bisimulations preserves formulas of modal logic.

III. Until and Since over general linear time

The language $L(U, S, U', S')$ is generated by the 2-place connectives U and S along with classical $\neg$ and $\wedge$ and the Stavi connectives, U' and S'. That is, we define the set of formulas recursively to contain the atoms and for formulas α and β we include $\neg\alpha$, $\alpha \wedge \beta$, $U(\alpha, \beta)$, $S(\alpha, \beta)$, $U'(\alpha, \beta$

1530-1311/13 $26.00 © 2013 IEEE
DOI 10.1109/TIME.2013.18

and $S'(\alpha, \beta)$. For the detailed syntax and semantics of this language, see [6].

This logic has been shown to be expressively complete for monadic first-order logic over arbitrary linear orders [6], generalizing the result of Kamp for Dedekind complete linear orders. This logic is decidable over arbitrary temporal structures and has been axiomatised [7].

IV. Building Structures

Model-checking and synthesis results require a description of a finite model, whereas the natural representation of general linear models is typically given over an infinite (and often uncountable) set of points. From [8] and [3], we know of a set of operations that are able to generate sufficiently complex structures for model-checking and synthesis purposes. These are described in the following section.

Our main artifact of study in this paper is a notation which allows us to describe temporal structures in sufficient detail to distinguish all concepts expressible in the monadic first-order theory of $<$. These structures are defined in terms of simple basic structures via a small number of ways of putting structures together to form larger ones.

The general idea is simple: using singleton structures (the flow of time is one point), we build up to more complex structures by the recursive application of four operations. The operations are:

- the *concatenation* of two structures, (placing one after the other) consisting of one followed by the other;
- ω repeats of a given structure *trailing* off into the future;
- ω repeats of a given structure trailing back into the past (or *leading* up to a point in the present)
- and making a densely thorough *shuffle* of copies from a finite set of structures.

These operations are well-known from the study of linear orders (see, for example, [8]).

We transform these operations into syntactic artefacts called *Linear Model Expressions*, which are an abstract syntax for defining models. Suppose that Σ is some alphabet (in the context of linear temporal logic, Σ would represent the set of propositional atoms true at a point. Linear model expressions are constructed using the follow set of primitive operators:

$$\mathcal{I} ::= \ a \mid \lambda \mid \mathcal{I} + \mathcal{J} \mid \overleftarrow{\mathcal{I}} \mid \overrightarrow{\mathcal{I}} \mid \langle \Gamma \rangle$$

where $a \in \Sigma$, for some alphabet Σ (in the context of linear temporal logic, Σ would represent the set of atoms true at a point), and Γ is a finite non-empty set of linear model expressions. We refer to these operators, respectively, as *a letter*, *the empty order*, *concatenation*, *lead*, *trail*, and *shuffle*. Let $E(\Sigma)$ be the set of all expressions generated over the alphabet Σ.

DEFINITION 2 (Correspondence): Given $\Sigma = 2^{\mathbf{L}}$, a model expression $\mathcal{I}$ *corresponds* to a structure as follows:

- λ is the empty sequence and corresponds to the frame $(\emptyset, <, h)$ where $<$ and h are empty relations.
- a corresponds to any single point model $(\{x\}, <, h)$ where $<$ is the empty relation and $h(p) = x$ if and only if $p \in a$.

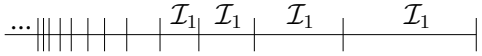

Fig. 1. The lead operation, where $\mathcal{I} = \overleftarrow{\mathcal{I}_1}$

For the inductive cases we require the notion of an isomorphism (Definition 1). Then:

- $\mathcal{I} + \mathcal{J}$ corresponds to a structure $(T, <, h)$ if and only if T is the disjoint union of two sets U and V, where $\forall u \in U$, $\forall v \in V$, $v < u$ (we write $U < V$, from here on) and $\mathcal{I}$ corresponds to $(U, <, h)$ and $\mathcal{J}$ corresponds to $(V, <, h)$.
- $\overleftarrow{\mathcal{I}}$ corresponds to the structure $(T, <, h)$ if and only if T is the disjoint union of sets $\{U_i | i \in \omega\}$ where for all i, $U_{i+1} < U_i$, and $\mathcal{I}$ corresponds to $(U_i, <, h)$.
- $\overrightarrow{\mathcal{I}}$ corresponds to the structure $(T, <, h)$ if and only if T is the disjoint union of sets $\{U_i | i \in \omega\}$ where for all i, $U_i < U_{i+1}$, and $\mathcal{I}$ corresponds to $(U_i, <, h)$.
- $\langle \Gamma \rangle$ corresponds to the structure $(T, <, h)$ if and only if T is the disjoint union of sets $\{U_i | i \in \mathbb{Q}\}$ where
  1) for all $i \in \mathbb{Q}$ $(U_i, <, h)$ corresponds to some $\gamma \in \Gamma$,
  2) for every $\gamma \in \Gamma$, for every $a \neq b \in \mathbb{Q}$, there is some $k \in (a, b)$ where γ corresponds to $(U_k, <, h)$,
  3) for every $a < b \in \mathbb{Q}$, $U_a < U_b$.

PROPOSITION 1: Every satisfiable formula of $L(U, S, U', S')$ is satisfied by a temporal structure that corresponds to a model expression.

This is proven in [9]. It provides a strong motivation for investigating model expressions, as they are able to finitely represent temporal structures satisfying logical formulae, which is important for such tasks as model-checking, synthesis and counter-example generation [10].

We will give an illustration of the non-trivial operations below. The *lead* operation, $\mathcal{I} = \overleftarrow{\mathcal{I}_1}$ corresponds ω submodels, each corresponding to $\mathcal{I}_1$, and each preceding the last, as illustrated in Figure 1.

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.

The model expression $\mathcal{I} = \langle \Gamma \rangle$ ($\mathcal{I}$ is the shuffle of Γ) corresponds to a dense, thorough mixture of intervals corresponding to the elements of Γ, without endpoints. We define the shuffle operation using the rationals, $\mathbb{Q}$ as they are a convenient linear order to describe a dense, thorough mixing of intervals.

The definition of model expressions is not deterministic, as the construct for the shuffle $\langle \mathcal{I}_1, \ldots, \mathcal{I}_n \rangle$ does not specify how the structures corresponding to $\mathcal{I}_1, \ldots, \mathcal{I}_n$ are mapped to the rationals. We show that this inconsequential, and as long as the mapping is dense for each i from 1 to n, the resulting structures will be isomorphic.

LEMMA 1: Suppose that $\mathcal{T} = (T, <, h)$ and $\mathcal{T}' = (T', <', h')$ both correspond to a model expression $\mathcal{I}$. Then $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic.

Proof: We prove this by induction over the complexity of model expressions. The base cases for letters and λ are trivial.

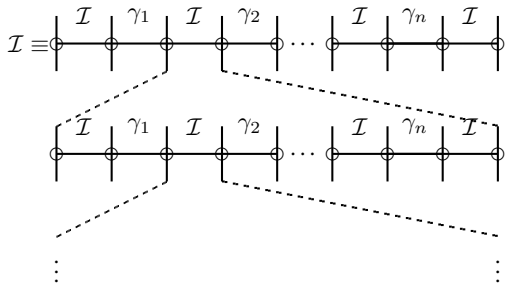

Fig. 2. The shuffle operation, where $\mathcal{I} = \langle\Gamma\rangle$, and $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$.

For concatenation, suppose two structures $\mathcal{T}$ and $\mathcal{T}'$ correspond to the model expression $\mathcal{I}_1 + \mathcal{I}_2$. Then $\mathcal{T} = (U \cup V, <, h)$ where $(U, <, h)$ corresponds to $\mathcal{I}_1$, $(V, <, h)$ corresponds to $\mathcal{I}_2$ and every element of U is less than every element of V. We also have $\mathcal{T}' = (U' \cup V', <', h')$ with similar constraints, so by the induction hypothesis $(U, <, h)$ is isomorphic to $(U', <', h')$ and likewise $(V, <, h)$ is isomorphic to $(V', <', h')$. Taking the union of these two isomorphisms we have an isomorphism from $\mathcal{T}$ to $\mathcal{T}'$ as required.

For trail, suppose two structures $\mathcal{T}$ and $\mathcal{T}'$ correspond to the model expression $\overrightarrow{\mathcal{I}_2}$. Then $\mathcal{T} = (\bigcup_{i\in\omega} U_i, <, h)$ where for all $i \in \omega$, $(U_i, <, h)$ corresponds to $\mathcal{I}_1$, and for all $i \in \omega$, every element of U_i is less than every element of U_{i+1}. We also have $\mathcal{T}' = (\bigcup_{i\in\omega} U'_i, <', h')$ with similar constraints, so by the induction hypothesis $(U_i, <, h)$ is isomorphic to $(U'_i, <', h')$ for every $i \in \omega$. Taking the union of these isomorphisms we have an isomorphism from $\mathcal{T}$ to $\mathcal{T}'$ as required.

The case for lead is similar so that leaves shuffles. Suppose two structures $\mathcal{T}$ and $\mathcal{T}'$ correspond to the model expression $\langle\Gamma\rangle$. From Definition 2 we have $\mathcal{T} = (\bigcup_{i\in\mathbb{Q}} U_i, <, h)$, and $\mathcal{T} = (\bigcup_{i\in\mathbb{Q}} U'_i, <', h')$ where the U_i and U'_i are disjoint, and correspond to the model expressions $\gamma \in \Gamma$. As the correspondence between the U_i and the model expressions is dense, we can build a 1-1 mapping $\pi \subset \mathbb{Q} \times \mathbb{Q}$ by a transfinite induction: we enumerate $\mathbb{Q}$ and process them in order to define the mappings π_x for $x \in \omega$. Suppose that i is the x^{th} element of the enumeration and U_i corresponds to $\gamma \in \Gamma$. If i is not already paired with any element in π_{x-1}, we let j be the least element greater than i where j is in the domain of π_{x-1}, and the let k be the greatest element less than i that appears in the domain of π_{x-1}. Then select some ℓ where $\pi_{x-1}(k) < \ell < \pi_{x-1}(j)$, and where U'_ℓ corresponds to γ. If no such j or k exists, we may ignore the corresponding constraint, and we can always find such an ℓ because the shuffle is dense). We can then repeat the process to find some ℓ' so that $U_{\ell'}$ is an isomorphic match for U'_i, and add both pairs, (i, ℓ) and (ℓ', i), to the mapping to define π_x. Note that the mapping is designed so that it is order-preserving (i.e. $pi_x(i) < \pi_x(j)$ only if $i < j$. The limit step takes the union of all interim mappings and will define a total one-to-one order-preserving mapping from $\mathbb{Q}$ to $\mathbb{Q}$ where U_i and $U'_{\pi(i)}$ correspond to the same model expression. By the induction hypothesis U_i and $U_{\pi(i)}$ must be isomorphic, and thus the union of these isomorphisms will be an isomorphism from $\mathcal{T}$ to $\mathcal{T}'$. ■

LEMMA 2: Suppose that $\mathcal{T} = (T, <, h)$ and $\mathcal{T}' = (T', <', h')$ are isomorphic. Then for any model expression $\mathcal{I}$ we have $\mathcal{T}$ corresponds to $\mathcal{I}$ if and only if $\mathcal{T}'$ corresponds to $\mathcal{I}$.

Proof: It is straightforward to see that the composition of an isomorphism and a correspondence will be a correspondence. ■

From these lemmas, we have the following definition.

DEFINITION 3: We say two model expressions $\mathcal{I}_1$ and $\mathcal{I}_2$ are *similar* (written $\mathcal{I}_1 \simeq \mathcal{I}_2$) if and only if there is some structure $\mathcal{T} = (T, <, h)$ such that both $\mathcal{I}_1$ and $\mathcal{I}_2$ correspond to $\mathcal{T}$.

Note that from Lemmas 2 and 1 it follows that *similarity* ($\simeq$) is an equivalence relation over the set of model expressions. Model expressions give us a grammar that describes linear sequences in a similar manner to the way regular expressions describe words over a given alphabet. However, there is an important difference in that while regular expressions contain non-deterministic operators (such as the Kleene star which can match any number of repetitions, or disjunctions), the interpretation of every operator in a model expression is deterministic up to isomorphism (Lemma 1). Therefore, the better analogue for a model expression is a single word over a finite alphabet. In future work we will examine an analogue of regular expressions for non-discrete time flows, such as a structure that admits general linear structures as traces. However, it is important to provide a solid foundation for the individual linear structures first, which is the focus of this paper.

We now focus on the question of equivalence for two model expressions. Definition 3 provides a semantic notion of what it means for two model expressions to represent the same information. The following section examines an algebraic approach for demonstrating that two model expressions represent the same information.

V. An algebra for model expressions

In this section we present an algebra for model expressions. The algebra is presented as a series of identities or equivalences with the intent that any two equivalent model expressions are isomorphic to exactly the same set of temporal structures. We write $\mathcal{I} \equiv \mathcal{J}$ to indicate that two model expressions, $\mathcal{I}$ and $\mathcal{J}$ are equivalent in the algebraic sense.

DEFINITION 4: The *model expression algebra* is as follows.

Associativity

$$1.\quad \mathcal{I}_1 + (\mathcal{I}_2 + \mathcal{I}_3) \equiv (\mathcal{I}_1 + \mathcal{I}_2) + \mathcal{I}_3$$

The shuffle identities

2. $\overrightarrow{\langle\Theta\rangle + \theta} \equiv \langle\Theta\rangle$ where $\theta \in \Theta$
3. $\overleftarrow{\theta + \langle\Theta\rangle} \equiv \langle\Theta\rangle$ where $\theta \in \Theta$
4. $\langle\Theta\rangle + \theta + \langle\Theta\rangle \equiv \langle\Theta\rangle$ where $\theta \in \Theta$
5. $\langle\Gamma \cup \{\theta + \langle\Theta\rangle + \theta'\}\rangle \equiv \langle\Theta\rangle$ where $\Gamma \subset \Theta, \theta, \theta' \in \Theta$

The lead/trail identities

6. $\overrightarrow{\mathcal{I}_1 + \mathcal{I}_2} \equiv \mathcal{I}_1 + \overrightarrow{\mathcal{I}_2 + \mathcal{I}_1}$
7. $\overrightarrow{\mathcal{I} + \ldots + \mathcal{I}} \equiv \overrightarrow{\mathcal{I}}$
8. $\overleftarrow{\mathcal{I}_1 + \mathcal{I}_2} \equiv \overleftarrow{\mathcal{I}_2 + \mathcal{I}_1} + \mathcal{I}_2$
9. $\overleftarrow{\mathcal{I} + \ldots + \mathcal{I}} \equiv \overleftarrow{\mathcal{I}}$

The λ identities

10. $\mathcal{I} + \lambda \equiv \mathcal{I}$
11. $\lambda + \mathcal{I} \equiv \mathcal{I}$
12. $\overrightarrow{\lambda} \equiv \lambda$
13. $\overleftarrow{\lambda} \equiv \lambda$
14. $\langle \lambda \rangle \equiv \lambda$
15. $\langle \Theta \rangle \equiv \langle \Theta \cup \{\lambda\} \rangle$

The substitution rule

$$\text{If } \mathcal{I} \equiv \mathcal{J} \text{ then } \mathcal{K} \equiv \mathcal{K}[\mathcal{I} \backslash \mathcal{J}]$$

where $\mathcal{K}[\mathcal{I} \backslash \mathcal{J}]$ is the expression $\mathcal{K}$ with all occurrences of the subexpression $\mathcal{I}$ replaced with the expression $\mathcal{J}$.

We note that this algebra is actually a schema due to identities (2-5, 7 and 9) which apply to a qualified set of terms. The identities (2-5) also refer to set operations which can further complicate derivations. For example, we can show $\langle \{\mathcal{I}, \mathcal{I} + \lambda\} \rangle$ is equivalent to $\langle \{\mathcal{I}\} \rangle$ for any model expression $\mathcal{I}$, by applying equivalence (10) to note that $\mathcal{I} + \lambda \equiv \mathcal{I}$, and then applying the substitution rule to show $\langle \{\mathcal{I}, \mathcal{I} + \lambda\} \rangle$ is equivalent to $\langle \{\mathcal{I}, \mathcal{I}\} \rangle$. At this point we note that $\{\mathcal{I}, \mathcal{I}\}$ is simply the set $\{\mathcal{I}\}$, but this final step is in some sense external to our system as it appeals to the definition of a set.

EXAMPLE 1: For an example of a more complex identity, suppose that $a \in \Sigma$ is some letter and consider the equivalence $\overleftarrow{a + \langle \{a + a\} \rangle + a} \equiv \langle \{a + a\} \rangle + a$. We can see that such an identity makes sense semantically. On the right side we have a dense mix of $\{a + a\}$ followed by a single point a. On the left we have an ω-sequence leading to the left of the point a, followed a dense mix of $a + a$, followed by a point a. When we lay this ω-sequence out we find the pattern:

$$\ldots + a + \langle \{a + a\} \rangle + a + a + \langle \{a + a\} \rangle + a + a + \langle \{a + a\} \rangle + a.$$

Now, each instance of $\langle \{a+a\} \rangle + a + a + \langle \{a+a\} \rangle$ is equivalent to $\langle \{a + a\} \rangle$, so the whole structure collapses down to $\langle \{a + a\} \rangle + a$, and hence the expressions are semantically equivalent. With respect to the algebra, we have the derivation:

$$\begin{aligned} \overleftarrow{a + \langle \{a + a\} \rangle + a} &\equiv \overleftarrow{a + a + \langle \{a + a\} \rangle} + a \qquad (8) \\ &\equiv \langle \{a + a\} \rangle + a \qquad (3) \end{aligned}$$

EXAMPLE 2: A useful derivation to have is the fixed-point definition of a lead: $\overleftarrow{\mathcal{I}} + \mathcal{I} \equiv \mathcal{I}$. We can show this with the following derivations:

$$\begin{aligned} \overleftarrow{\mathcal{I}} &\equiv \overleftarrow{\lambda + \mathcal{I}} \qquad \text{(11) and substitution} \\ &\equiv \overleftarrow{\mathcal{I} + \lambda} + \mathcal{I} \qquad (8) \\ &\equiv \overleftarrow{\mathcal{I}} + \mathcal{I} \qquad \text{(10) and substitution.} \end{aligned}$$

In the next few sections we will show that the algebra is sound (so that $\mathcal{I}_1 \equiv \mathcal{I}_2$ *only if* $\mathcal{I}_1 \simeq \mathcal{I}_2$) and complete (so that $\mathcal{I}_1 \equiv \mathcal{I}_2$ *if* $\mathcal{I}_1 \simeq \mathcal{I}_2$).

VI. Soundness

We must first convince ourselves that the model expression algebra will only generate valid equivalences.

LEMMA 3 (Soundness): Suppose that $\mathcal{I}_1$ and $\mathcal{I}_2$ are model expressions such that $\mathcal{I}_1 \equiv \mathcal{I}_2$. Then $\mathcal{I}_1 \simeq \mathcal{I}_2$

Proof: (Sketch)

It is sufficient to show that all rules preserve equivalence with respect to the correspondence relation, so that is $\mathcal{I}$ corresponds to $\mathcal{T}$ and $\mathcal{I} \equiv \mathcal{J}$ is an identity of the model expression algebra, then $\mathcal{J}$ also corresponds to $\mathcal{T}$. As equivalence ($\equiv$) is simply defined as the result of the iterated application of these rules, it follows that equivalence preserves the correspondence relation. Showing that each rule preserves the correspondence relation is straightforward and we present brief arguments for some of the equivalences below:

1 $\mathcal{I}_1 + (\mathcal{I}_2 + \mathcal{I}_3) \equiv (\mathcal{I}_1 + \mathcal{I}_2) + \mathcal{I}_3$: the concatenation operator is clearly associative.

2 $\overrightarrow{\langle \Theta \rangle + \theta} \equiv \langle \Theta \rangle$ where $\theta \in \Theta$: an ω-sequence of shuffles is isomorphic to a shuffle.

4 $\langle \Theta \rangle + \theta + \langle \Theta \rangle \equiv \langle \Theta \rangle$ where $\theta \in \Theta$: this is the essential definition of a shuffle operator.

5 $\langle \Gamma \cup \{\theta + \langle \Theta \rangle + \theta'\} \rangle \equiv \langle \Theta \rangle$ where $\Gamma \subset \Theta$ and $\theta, \theta' \in \Theta$: if Γ is a subset of Θ then $\langle \Gamma \cup \{\langle \Theta \rangle\} \rangle$ is a dense mix of elements from Γ and $\langle \Theta \rangle$. However, as elements of Γ already appear densely in $\langle \Theta \rangle$, so nothing is gained from having them appear independently of the elements in Θ. Likewise, within a shuffle, there is effectively no difference between a closed interval and an open interval, so the addition of θ, θ' at the ends of $\langle \Theta \rangle$ are negligible.

6 $\overrightarrow{\mathcal{I}_1 + \mathcal{I}_2} \equiv \mathcal{I}_1 + \overrightarrow{\mathcal{I}_2 + \mathcal{I}_1}$: an ω-sequence alternating between $\mathcal{I}_1$ and $\mathcal{I}_2$ is the same as an ω alternating between $\mathcal{I}_2$ and $\mathcal{I}_1$, with $\mathcal{I}_1$ appended to the front.

7 $\overrightarrow{\mathcal{I} + \ldots + \mathcal{I}} \equiv \overrightarrow{\mathcal{I}}$: an ω-sequence of a finite homogeneous sequence of $\mathcal{I}$ is just an ω-sequence of $\mathcal{I}$.

10-15 These rules capture how the empty sequence λ acts as a kind of identity for all operators.

Finally we note that the substitution rule must be sound since any substructure that $\mathcal{I}$ corresponds to must also be a substructure that $\mathcal{J}$ corresponds to, so the correspondence of $\mathcal{K}$ to a structure will not be affected by the substitution. ■

VII. Completeness

Completeness requires us to show that any two expressions that correspond to a common temporal structure, are provably equivalent using the model expression algebra.

LEMMA 4 (Completeness): Suppose that $\mathcal{I}$ and $\mathcal{J}$ are model expressions such that $\mathcal{I} \simeq \mathcal{J}$. Then $\mathcal{I} \equiv \mathcal{J}$.

This is the more complicated lemma to prove and will rely on a sequence of definitions and sub-lemmas. We will give these below and bring them together in the proof of Theorem 1. Our proof uses a two player game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ derived from the two model expressions $\mathcal{I}$ and $\mathcal{J}$. The game is such that player *Felix* (the *Duplicator*) has a winning strategy if and only if $\mathcal{I} \simeq \mathcal{J}$, and otherwise player *Ralph* (the *Spoiler*) has a winning strategy. Then, from *Felix*'s winning strategy (and

Ralph's lack of a winning strategy) we are able to extract a series of equivalences showing $\mathcal{I} \equiv \mathcal{J}$.

DEFINITION 5: Given two model expressions $\mathcal{I}$ and $\mathcal{J}$ we define $\mathcal{G}(\mathcal{I}, \mathcal{J})$ to be the *equivalence game of $\mathcal{I}$ and $\mathcal{J}$*, where:

1) The game is played between two players, *Felix* and *Ralph*.
2) At any time, the state of game is a pair $(\mathcal{I}', \mathcal{J}')$ where $\mathcal{I}'$ and $\mathcal{J}'$ are model expressions, and the game starts at the state $(\mathcal{I}, \mathcal{J})$.
3) Given the state $(\mathcal{I}', \mathcal{J}')$, the next step of the game proceeds as follows:
   a) *Ralph* performs a sequence of $\equiv$ transformations to $\mathcal{I}'$ manipulating it into an expression $\mathcal{I}'_1 + \mathcal{I}'_2$
   b) *Felix* performs a sequence of $\equiv$ transformations to $\mathcal{J}'$ manipulating it into an expression $\mathcal{J}'_1 + \mathcal{J}'_2$
   c) *Ralph* then selects the next state of the game to be either $(\mathcal{I}'_1, \mathcal{J}'_1)$ (the *left*), $(\mathcal{J}'_1, \mathcal{I}'_1)$ (the *left switch*), $(\mathcal{I}'_2, \mathcal{J}'_2)$ (the *right*), or $(\mathcal{J}'_2, \mathcal{I}'_2)$ (the *right switch*).
4) *Ralph* wins the game if it reaches a state
   a) $(\lambda, \mathcal{I})$ where $\mathcal{I} \neq \lambda$,
   b) $(a, \mathcal{I})$ where a is an atom and $\mathcal{I} \neq a$,

   Otherwise *Felix* wins.

Note that the game is not necessarily finite, but the only way *Ralph* can win is to force the game into a final state in a finite number of moves.

LEMMA 5: For all model expressions $\mathcal{I}$ and $\mathcal{J}$, the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ is determined. That is, either *Ralph* or *Felix* has a winning strategy.

Proof: This follows from the fact that there are only a countable number of winning plays for *Ralph*, and they are all finite. Therefore, an inductive construction of *Ralph*'s winning set has a well-defined limit. ■

EXAMPLE 3: For an example of the game, consider Example 1. Here it is clear that *Felix* has a winning strategy. Let $\mathcal{I} = \overleftarrow{a + \langle\{a + a\}\rangle + a}$ and $\mathcal{J} = \langle\{a + a\}\rangle + a$. From the game state $(\mathcal{I}, \mathcal{J})$, *Ralph* will move first and produce a derivation $\mathcal{I} \equiv \mathcal{I} + \mathcal{I}''$. *Felix*'s winning strategy is simply to first repeat the derivation in Example 1 and compose this derivation with which ever derivation *Ralph* produced, so we have $\mathcal{J} \equiv \mathcal{I} + \mathcal{I}''$. From then he can mimic exactly every move *Ralph* makes. *Ralph* could reach states (a, a) or (λ, λ) but neither of these states are winning, so the game continues indefinitely (because the λ-identities always ensure another move is available).

For a game in which *Ralph* has a winning strategy, suppose that $\mathcal{I}$ is defined as above, but $\mathcal{J} = \langle\{a + a\}\rangle$. In this case *Ralph* could choose the derivation $\mathcal{I} \equiv \langle\{a + a\}\rangle + a$ that appeared in Example 1. Whatever derivation $\mathcal{J} = \mathcal{J}' + \mathcal{J}''$ *Felix* tries, he will always find that $\mathcal{J}''$ contains a dense mix of $\{a + a\}$ or λ and is definitely not equal to a. So *Ralph* wins the game by choosing the right.

We will show that a winning strategy for *Felix* in the game $\mathcal{G}(M_1, M_2)$ is equivalent to both $M_1 \simeq M_2$ and $M_1 \equiv M_2$.

We first prove the following sub-lemma, by induction over the complexity of formulas.

LEMMA 6: *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ if and only if $\mathcal{I} \simeq \mathcal{J}$.

Sketch: If $\mathcal{I} \simeq \mathcal{J}$ then both expressions correspond to a common linear structure, $\mathcal{T} = (T, <, h)$. We will use this correspondence to guide a winning strategy for *Felix* in the game. We will proceed by induction and we will suppose that we have game states $(\mathcal{I}^i, \mathcal{J}^i)$ where both $\mathcal{I}^i$ and $\mathcal{J}^i$ correspond to a structure $\mathcal{T}^i$. The base of our induction is $\mathcal{I}^0 = \mathcal{I}$, $\mathcal{J}^0 = \mathcal{J}$ and $\mathcal{T}^0 = \mathcal{T}$. Now suppose that given a state $(\mathcal{I}^i, \mathcal{J}^i)$, *Ralph* plays a move $\mathcal{I}^i \equiv \mathcal{I}_1 + \mathcal{I}_2$. From Lemma 3 we must have that $\mathcal{I}_1 + \mathcal{I}_2$ also corresponds to $\mathcal{T}^i = (T^i, <^i, h^i)$. Applying Definition 2 this means that T^i is the disjoint union of two sets U and V where $V < U$ and $\mathcal{I}_1$ corresponds to $(U, <, h)$ and $\mathcal{I}_2$ corresponds to $(V, <, h)$. As $\mathcal{J}^i$ also corresponds to $\mathcal{T}^i$ we need to find a derivation $\mathcal{J}^i \equiv \mathcal{J}_1 + \mathcal{J}_2$ where $\mathcal{J}_1$ corresponds to $(U, <, h)$ and $\mathcal{J}_2$ corresponds to $(V, <, h)$. We can then show that as long as *Felix* continues to produce such derivations, then *Ralph* can never reach a winning state. We will provide an inductive argument to show the *Felix* can always find a suitable derivation (and the previous two cases form the basis of this induction). The claim is that given an expression $\mathcal{K}$ that corresponds to a structure, $\mathcal{T} = (T, <, h)$, and a partition of that structure into two sets U and V where $V < U$, we can always find a derivation $\mathcal{K} \equiv \mathcal{K}_1 + \mathcal{K}_2$ such that $\mathcal{K}_1$ corresponds to $(U, <, h)$ and $\mathcal{K}_2$ corresponds to $(V, <, h)$. We proceed by induction over the complexity of formulas, where the first two cases are the basis, and the inductive steps follow:

- If $\mathcal{K} = \lambda$, then $T = \emptyset$ and thus U and V are also the empty set. *Felix* can produce the derivation $\mathcal{K} \equiv \lambda + \lambda$ using (11).
- If $\mathcal{K} = a$, where $a \in \Sigma$, then it must be the T is a single point, x, where $h^i(x) = a$. As $\mathcal{I}^i_1 + \mathcal{I}^i_2$ corresponds to $\mathcal{T}$ we must have either $U = \{x\}$ and $V = \emptyset$ for $U = \emptyset$ and $V = \{x\}$. In the first case *Felix* can produce the derivation $\mathcal{K} = a + \lambda$ (11) and in the second case *Felix* can produce the derivation $\mathcal{K} = \lambda + a$ (12).
- If $\mathcal{K} = \mathcal{K}_1 + \mathcal{K}_2$ then T^i is the disjoint union of two sets U' and V' where for all $u \in U'$, for all $v \in V'$, $u < v$. Suppose without loss of generality $V' \subset V$. Let $V^* = V \cap U'$. Then $\mathcal{K}_1$ corresponds to $(U \cup V^*, <, h)$ where everything in U is less than everything in V^*. Applying the induction hypothesis $K_1 \equiv \mathcal{K}_3 + \mathcal{K}_4$ where K_2 corresponds to U and $\mathcal{K}_4$ corresponds to V^*. Applying the substitution rule we have $\mathcal{K} \equiv (\mathcal{K}_3 + \mathcal{K}_4) + \mathcal{K}_2$ and by (1), we can complete the derivation $\mathcal{K} \equiv \mathcal{K}_3 + (\mathcal{K}_4 + \mathcal{K}_5)$ where $\mathcal{K}_3$ corresponds to U and $\mathcal{K}_4 + \mathcal{K}_5$ corresponds to V.
- If $\mathcal{K} = \overleftarrow{\mathcal{K}_1}$ then by Definition 2, T is the disjoint union of sets $\{U_i | i \in \omega\}$ where for all i, for all $u \in U_i$, for all $v \in U_{i+1}$, $v < u$, and $\mathcal{K}_1$ corresponds to $(U_i, <, h)$. As T is totally ordered by $<$ there is some greatest $j \in \omega$ such that $\bigcup_{i<j} U_i \subseteq V$. We can apply the derivation of Example 2 j times so that $\mathcal{K} \equiv \overleftarrow{\mathcal{K}_1} + j\mathcal{K}_1$. Let $U^* = U \cap U_j$, $V^* = V \cap U_j$. Then $\mathcal{K}_1$ corresponds to $(U^* \cup V^*, <, h)$ where everything in U^* is less than everything in V^*. Applying the inductive hypothesis we have $\mathcal{K}_1 \equiv \mathcal{K}_2 + \mathcal{K}_3$ where $\mathcal{K}_2$ corresponds to U^* and $\mathcal{K}_3$ corresponds to V^*. Using the substitution rule and (1), we can show

that $\mathcal{K} \equiv \overleftarrow{\mathcal{K}_1} + \mathcal{K}_2 + \mathcal{K}_3 + (j-1).\mathcal{K}_1$ where $\overleftarrow{\mathcal{K}_1} + \mathcal{K}_2$ corresponds to U and $\mathcal{K}_3 + (j-1).\mathcal{K}_1$ corresponds to V.
- The case for $\mathcal{K} = \overrightarrow{\mathcal{K}_1}$ is the symmetric case to $\mathcal{K} = \overleftarrow{\mathcal{K}_1}$.
- If $\mathcal{K} = \langle\Gamma\rangle$ where $\Gamma = \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$ then T is the disjoint union of sets $\{U_i | i \in \mathbb{Q}\}$ satisfying the conditions laid out in Definition 2. Let i be the element of $\mathbb{Q}$ such that $U_i \not\subset U$ and $U_i \not\subset V$, if it exists (i.e. U_i is the substructure that crosses from one partition into the other). If such an element doesn't exist, then every U_i is either entirely in U or entirely in V and there are three cases to consider.
  1) If there is a least $U_i \subset V$ (with respect to $<$) such that $\mathcal{I}_j$ corresponds to U_i, we have by (4), $\mathcal{K} \equiv \langle\Gamma\rangle + \mathcal{I}_j + \langle\Gamma\rangle$, where $\langle\Gamma\rangle$ corresponds to U and $\mathcal{I}_j + \langle\Gamma\rangle$ corresponds to V.
  2) If there is a greatest $U_i \subset U$ such that $\mathcal{I}_j$ corresponds to U_i, we have by (4), $\mathcal{K} \equiv \langle\Gamma\rangle + \mathcal{I}_j + \langle\Gamma\rangle$, where $\langle\Gamma\rangle + \mathcal{I}_j$ corresponds to U and $\langle\Gamma\rangle$ corresponds to V.
  3) If there is neither a greatest $U_i \subset U$, nor a least $U_i \subset V$, then by (15) $\mathcal{K} \equiv \langle\Gamma \cup \{\lambda\}\rangle$ and by (4) and (15) $\mathcal{K} \equiv \langle\Gamma\rangle + \lambda + \langle\Gamma\rangle$ and finally by (12) $\mathcal{K} \equiv \langle\Gamma\rangle + \langle\Gamma\rangle$ where $\langle\Gamma\rangle$ corresponds to both U and V.

  Otherwise there is an element $i \in Q$ such that $U_i \not\subset U$ and $U_i \not\subset V$. Let $U^* = U \cap U_i$, $V^* = V \cap U_i$ and suppose that $\mathcal{I}_j$ corresponds to U_i. By the induction hypothesis we have $\mathcal{I}_j \equiv \mathcal{K}_1 + \mathcal{K}_2$ where $\mathcal{K}_1$ corresponds to U^* and $\mathcal{K}_2$ corresponds to V^*. By (14) $\mathcal{K} \equiv \langle\Gamma\rangle + \mathcal{I}_j + \langle\Gamma\rangle$, and applying the substitution rule we have $\mathcal{K} \equiv \langle\Gamma\rangle + \mathcal{K}_1 + \mathcal{K}_2 + \langle\Gamma\rangle$ where $\langle\Gamma\rangle + \mathcal{K}_1$ corresponds to U and $\mathcal{K}_2 + \langle\Gamma\rangle$ corresponds to V.

Therefore, what ever choice *Ralph* makes, *Felix* can always respond finding a pair $\mathcal{J}_1, \mathcal{J}_2$ corresponding to the respective partition of T. The induction shows that *Ralph* will never be able to reach a winning state, so it must be a winning strategy for *Felix*.

Conversely, if *Felix* has a winning strategy in the game, we can show that the two expressions correspond to isomorphic structures by building a model extracted from all plays of the game according to *Felix*'s winning strategy.

We define a non-deterministic strategy for *Ralph*. Since *Felix* has a winning strategy, *Felix* can win every game against this strategy. However, this strategy is defined so that every play that reaches an atomic state ((a, a), since it is a winning state for *Felix*), and this corresponds to a unique point in a linear structure $\mathcal{T}$. We will show the structure $\mathcal{T}$ corresponds to both $\mathcal{I}$ and $\mathcal{J}$ as required.

The strategy for *Ralph*, σ maps game states $(\mathcal{I}, \mathcal{J})$ to a derivation for $\mathcal{I}$, $\sigma(\mathcal{I}) = \mathcal{I}' + \mathcal{I}''$ where $\mathcal{I} \equiv \mathcal{I}' + \mathcal{I}''$. When $\mathcal{I}' + \mathcal{I}''$ is played, Player *Felix* will respond by applying the winning strategy to select a derivation $\mathcal{J} = \mathcal{J}' + \mathcal{J}''$. Then *Ralph* non-deterministically selects the left $(\mathcal{I}', \mathcal{J}')$ or the right $(\mathcal{I}'', \mathcal{J}'')$. The function σ is defined inductively over the complexity of expressions:

$$\sigma(\lambda) = \lambda + \lambda \qquad \sigma(a) = \textit{undefined}$$
$$\sigma(\mathcal{K} + \mathcal{K}') = \mathcal{K} + \mathcal{K}' \qquad \sigma(\overleftarrow{\mathcal{K}}) = \overleftarrow{\mathcal{K}} + \mathcal{K}$$
$$\sigma(\overrightarrow{\mathcal{K}}) = \mathcal{K} + \overrightarrow{\mathcal{K}} \qquad \sigma(\langle\Gamma\rangle) = (\langle\Gamma\rangle + \gamma) + \langle\Gamma\rangle$$

In the last clause γ is non-deterministically chosen from $\Gamma \cup \{\lambda\}$.

We consider all plays that may result from this non-deterministic strategy for *Ralph* against *Felix*'s winning strategy. Let $\Pi^{\mathcal{I}}_{\mathcal{J}}$ be the set of finite plays of game, $\pi = \pi_0\pi_1 \ldots \pi_n$ where $\pi_0 = (\mathcal{I}, \mathcal{J})$ and $\pi_n = (a, a)$ for some $a \in \Sigma$. We define an ordering over $\Pi^{\mathcal{I}}_{\mathcal{J}}$ by $\pi < \tau$ if and only if, for the largest j where for all $i < j$, $\pi_i = \tau_i$, we have π_j is a left move and τ_j is a right move. Finally, we define the function $h\colon \mathbf{L} \to \mathbf{2}^{\mathbf{\Pi}^{\mathcal{I}}_{\mathcal{J}}}$, by $\pi \in h(a)$ if and only if $\pi_n = (a, a)$. Note that there are also plays winning for *Felix* that end in the state (λ, λ), but these may be ignored as they correspond to the empty flow of time, and do not affect the final structure.

We can then show that $(\Pi^{\mathcal{I}}_{\mathcal{J}}, <, h)$ corresponds to both $\mathcal{I}$ and $\mathcal{J}$, by induction over the complexity of $\mathcal{I}$. The induction hypothesis is that for all sub-expressions $\mathcal{I}'$ of $\mathcal{I}$, for all $\mathcal{J}'$, if *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}', \mathcal{J}')$ then $(\Pi^{\mathcal{I}'}_{\mathcal{J}'}, <, h)$ corresponds to both $\mathcal{I}'$ and $\mathcal{J}'$. The base case of $\mathcal{I} = \mathcal{J} = a$ is obvious. The operators concatenation, lead, trail and shuffle are simple applications of the recursive definition of correspondence, given the soundness of the model expression algebra (Lemma 3). Therefore $\mathcal{I} \simeq \mathcal{J}$ as required. ■

The final part of the completeness proof will require us to show that *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ if and only if $\mathcal{I} \equiv \mathcal{J}$. The proof will proceed via induction over the complexity of model expressions, where the complexity of a model expression is given by the following definition.

DEFINITION 6: Model Expression Complexity We say one expression $\mathcal{I}$ is an *equivalent subexpression* of another $\mathcal{J}$ (written $\mathcal{I} \sqsubseteq \mathcal{J}$) if there exists expressions $\mathcal{K}$ and $\mathcal{K}'$ such that $\mathcal{K} + \mathcal{I} + \mathcal{K}' \equiv \mathcal{J}$.

The *complexity of a model expression* $\mathcal{I}$, is a map $c\colon E(\Sigma) \longrightarrow \omega \times \omega$ defined inductively over the formulaic complexity of model expressions as follows:

- **Atoms**
  - $\mathrm{c}(\lambda) = (0, 0)$
  - $\mathrm{c}(a) = (0, 1)$
- **Concatenation**
  - $\mathrm{c}(\mathcal{I} + \mathcal{I}') = \max(c(\mathcal{I}), c(\mathcal{I}'))$.
- **Lead/Trail**
  - If $\mathrm{c}(\mathcal{I}) = (a, b)$ and $\overleftarrow{\mathcal{I}} \sqsubseteq \mathcal{I}$, $\mathrm{c}(\overleftarrow{\mathcal{I}}) = (a, b)$.
  - If $\mathrm{c}(\mathcal{I}) = (a, b)$ and $\overleftarrow{\mathcal{I}} \not\sqsubseteq \mathcal{I}$, $\mathrm{c}(\overleftarrow{\mathcal{I}}) = (a, b+1)$.
  - If $\mathrm{c}(\mathcal{I}) = (a, b)$ and $\overrightarrow{\mathcal{I}} \sqsubseteq \mathcal{I}$, $\mathrm{c}(\overrightarrow{\mathcal{I}}) = (a, b)$.
  - If $\mathrm{c}(\mathcal{I}) = (a, b)$ and $\overrightarrow{\mathcal{I}} \not\sqsubseteq \mathcal{I}$, $\mathrm{c}(\overrightarrow{\mathcal{I}}) = (a, b+1)$.
- **Shuffle**
  - If for every $\gamma \in \Gamma$ we have $\langle\Gamma\rangle \sqsubseteq \gamma$, then $\mathrm{c}(\langle\Gamma\rangle) = \max\{\mathrm{c}(\gamma) \mid \gamma \in \Gamma\}$.
  - Otherwise if $\max\{\mathrm{c}(\gamma) \mid \gamma \in \Gamma\} = (a, b)$ then $\mathrm{c}(\langle\Gamma\rangle) = (a+1, 0)$.

We note that over $\omega \times \omega$, max is determined with respect to the lexicographical relation, so that $(a, b) < (c, d)$ if and only if $a < c$ or $a = c$ and $b < d$.

This complexity orders expressions in two dimensions. Essentially every nesting of a lead or a trail increases the complexity by one in the second dimension, and every nesting

of a shuffle increase the dimension by one in the first dimension and resets the second dimension to zero. Therefore every expression with a shuffle as the outermost operator will always have complexity $(a, 0)$. There are a few cases where leads, trails and shuffles of expressions don't substantially change the expression (see the shuffle identities in Definition 4), and in these case the complexity is left at $(a, 0)$.

Some examples of model expression complexities are:

$$\mathrm{c}(\overleftarrow{a}) = (0, 2) \quad (1)$$

$$\mathrm{c}(\langle\{a\}\rangle) = (1, 0) \quad (2)$$

$$\mathrm{c}(a + \overleftarrow{\langle\{a\}\rangle}) = (1, 0) \quad (3)$$

$$\mathrm{c}(\overleftarrow{a + \langle\{a\}\rangle}) = (1, 0) \quad (4)$$

$$\mathrm{c}(\overleftarrow{b + \langle\{a\}\rangle}) = (1, 1) \quad (5)$$

These formulas give some idea of how the complexities build. Each shuffle acts as a type of limit for the inductive application of lead and trails. We see in the formula (3) and (4) the complexity does not increase with the application of concatenation or lead operations, as these are "absorbed" by the underlying shuffle (in fact we can see that the formula in (4) is equivalent to the formula in (2). However, for (5) we see the complexity does increase because the sub-expression b cannot be absorbed by the shuffle (since the first clause for the lead complexity does not hold: $\overleftarrow{b + \langle\{a\}\rangle} \not\sqsubseteq b + \langle\{a\}\rangle$). In this case we say the subexpression b is an *impurity* in the expression $b + \langle\{a\}\rangle$.

LEMMA 7: If $\mathcal{I} \equiv \mathcal{J}$ then $\mathrm{c}(\mathcal{I}) = \mathrm{c}(\mathcal{J})$.

Proof: The result follows clearly from inspection of Definition 4. As the model expression complexity of the left and right side of each equivalence are clearly equal, it follows by a simple inductive argument that equivalent expressions must have the same model expression complexity. ■

LEMMA 8: If $\mathcal{I} \simeq \mathcal{J}$ then $\mathrm{c}(\mathcal{I}) = \mathrm{c}(\mathcal{J})$.

Proof: We give this proof by induction over the syntax of expressions. The base cases of λ and atoms, $a \in \Sigma$ are straight forward: if $a \simeq \mathcal{J}$ then $\mathcal{J}$ can only be an expression $\mathcal{K} + a + \mathcal{K}'$ where $\mathcal{K}$ and $\mathcal{K}'$ are expressions built only from λ, and as such both expressions must have complexity $(0, 1)$. The case for λ is also trivial.

For the inductive steps we consider the model expression $\mathcal{I}$, and suppose that both $\mathcal{I}$ and $\mathcal{J}$ correspond to some structure $\mathcal{T} = (T, <, h)$.

- Suppose $\mathcal{I} = \mathcal{I}' + \mathcal{I}''$ and $\mathcal{I} \simeq \mathcal{J}$. Therefore $\mathcal{J}$ also corresponds to $\mathcal{T}$ and there are subsets $U_1 < U_2$ of T such that $\mathcal{I}'$ corresponds to $(U_1, <, h)$ and $\mathcal{I}''$ corresponds to $(U_2, <, h)$. From Lemma 6, we know that $\mathcal{J} \equiv \mathcal{J}' + \mathcal{J}''$, where $\mathcal{J}'$ corresponds to $(U_1, <, h)$ and $\mathcal{J}''$ corresponds to $(U_2, <, h)$. By the induction hypothesis, $\mathrm{c}(\mathcal{I}') = \mathrm{c}(\mathcal{J}')$ and $\mathrm{c}(\mathcal{I}'') = \mathrm{c}(\mathcal{J}'')$ so the result follows from Definition 6.
- Suppose that $\mathcal{I} = \overleftarrow{\mathcal{I}'}$ and $\mathcal{I} \simeq \mathcal{J}$. Therefore there is some partition $\{U_i | i \in \omega,\ U_{i+1} < U_i\}$ of T such that for every $i \in \omega$, $\mathcal{I}'$ corresponds to U_i. There are two sub-cases to consider. If each $\mathcal{I}$ also corresponds to each U_i, then $\mathrm{c}(\mathcal{I}) = (a, 0)$ for some a, and the case may be treated as a shuffle below. Otherwise we will have $\mathrm{c}(\mathcal{I}) = (a, b)$ for some $b > 0$. Applying the same process from the previous case, for arbitrary n we may show that $\mathcal{J} \equiv \mathcal{J}_n + \ldots + \mathcal{J}_0$ where for all $i < n$, $\mathcal{J}_i$ corresponds to $(U_i, <, h)$. The only operations that can generate such a sequence are shuffle or lead, and as a shuffle has been ruled out we must have $\mathcal{J} \equiv \overleftarrow{\mathcal{J}'}$ for some $\mathcal{J}'$ that corresponds to $(U_0, <, h)$. By the inductive hypothesis $\mathrm{c}(\mathcal{I}') = (a, b) = \mathrm{c}(\mathcal{J}')$ for some (a, b) and hence $\mathrm{c}(\mathcal{I}) = \mathrm{c}(\mathcal{J}) = (a, b + 1)$.
- Suppose that $\mathcal{I} = \overrightarrow{\mathcal{I}'}$ and $\mathcal{I} \simeq \mathcal{J}$. This case is simply the mirror image of the previous case.
- Suppose that $\mathcal{I} = \langle\Gamma\rangle$ and $\mathcal{I} \simeq \mathcal{J}$. We can find a partition of T, $\{U_i | i \in \mathbb{Q}\}$ such that for each $a < b \in \mathbb{Q}$, $U_a < U_b$, and for each $\gamma \in \Gamma$ there is some c where $a < c < b$, and γ corresponds to $(U_c, <, h)$. Again, the processes of Lemma 6 can be applied to show that $\mathcal{J}$ must be equivalent to some shuffle $\langle\Theta\rangle$, where for every $a < b \in \mathbb{Q}$, for every $\theta \in \Theta$, there is some c where $a < c < b$ and θ corresponds to $(U_c, <, h)$. By the induction hypothesis for every $\gamma \in \Gamma$ there must be some $\theta \in \Theta$ such that $\mathrm{c}(\gamma) = \mathrm{c}(\theta)$ and vice versa. Therefore, we must have $\mathrm{c}(\mathcal{I}) = \mathrm{c}(\mathcal{J})$.

By induction over the syntax of expressions the Lemma must hold. ■

We note that as a corollary of Lemmas 6 and 8, if two expressions $\mathcal{I}$ and $\mathcal{J}$ have different model expression complexity, then *Ralph* must have a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$.

LEMMA 9: *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ if and only if $\mathcal{I} \equiv \mathcal{J}$.

Proof: If $\mathcal{I} \equiv \mathcal{J}$ then *Felix* has a clear winning strategy: his first move is to perform the deductive steps to transform $\mathcal{J}$ into $\mathcal{I}$ and append any deductions the *Ralph* made in his first move. From then on he simply mirrors every one of *Ralph*'s moves, guaranteeing a winning strategy.

For the converse, if *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ each move of *Felix* is going to involve a derivation. We can use this winning strategy to construct a complete derivation of the equivalence $\mathcal{I} \equiv \mathcal{J}$, proceeding by induction over the complexity of formulas as defined in the Definition 6. The induction hypothesis is that *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}', \mathcal{J}')$, for all $\mathcal{I}'$ with a model expression complexity less than $\mathrm{c}(\mathcal{I})$, and also for all $\mathcal{I}'$ which are strict subexpressions of $\mathcal{I}$.

The base of this induction will be expressions of complexity less than or equal to $(0, 1)$.

- If $\mathrm{c}(\mathcal{I}) = (0, 0)$ then it is clear from Definition 6 that $\mathcal{I} \equiv \lambda$. Therefore if *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ then it must be the case that $\mathcal{J} \equiv \lambda$, otherwise *Ralph* would have a winning move of $\mathcal{I} \equiv \lambda + \lambda$.
- If $\mathrm{c}(\mathcal{I}) = (0, 1)$ then from Definition 6 it follows that $\mathcal{I}$ is equivalent to a concatenation of atoms and λ (as any instance of lead, trail or shuffle applied to an atom would increase the complexity beyond $(0, 1)$. If *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ it must follow that for every decomposition $\mathcal{I} \equiv \mathcal{I}' + \mathcal{I}''$, *Felix* can find some decomposition $\mathcal{J} \equiv \mathcal{J}' + \mathcal{J}''$ such that *Felix*

has a winning strategy in the left, right, left switch and right switch states. The result follows by induction over the complexity of expressions (the standard definition, as opposed to the model expression complexity). The base of this induction has $\mathcal{I}$ and $\mathcal{J}$ as the same atom (since *Ralph* cannot win) and for the inductive step we may assume that $\mathcal{I}' \equiv \mathcal{J}'$ and $\mathcal{I}'' \equiv \mathcal{J}''$, so the result follows immediately.

There are now a number of different inductive cases to consider:

- $\mathcal{I} = \mathcal{I}' + \mathcal{I}''$ In this case it is clear that as *Felix* has a winning strategy, if *Ralph* plays $\mathcal{I}' + \mathcal{I}''$ then *Felix* can respond with a derivation $\mathcal{J} \equiv \mathcal{J}' + \mathcal{J}''$. As it is a winning strategy, this means that *Felix* will win both games $\mathcal{G}(\mathcal{I}', \mathcal{J}')$ and $\mathcal{G}(\mathcal{I}'', \mathcal{J}'')$. By the induction hypothesis we must have $\mathcal{I}' \equiv \mathcal{J}'$ and $\mathcal{I}'' \equiv \mathcal{J}''$, so by the substitution rule it follows that $\mathcal{I} \equiv \mathcal{J}$ as required.
- $\mathcal{I} = \overleftarrow{\mathcal{I}'}$. Suppose that for some $\mathcal{J}$, *Felix* has a winning strategy in the $\mathcal{G}(\mathcal{I}, \mathcal{J})$. From this we can deduce that $\mathrm{c}(\mathcal{I}) = \mathrm{c}(\mathcal{J})$. Furthermore, we may suppose that $\mathrm{c}(\mathcal{I}) = (a, b)$ where $b > 0$, otherwise we would have $\mathcal{I}$ being equivalent to a shuffle operation, and covered by the case below. Given that $\mathrm{c}(\mathcal{J}) = (a, b)$ for $b > 0$ and *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ we can see that $\mathcal{J}$ must be of the form $\overleftarrow{\mathcal{J}'} + \mathcal{J}''$ where $\mathrm{c}(\mathcal{J}'') \leq \mathrm{c}(\mathcal{J}')$. (If the most complex subexpression of $\mathcal{J}$ was a trail, *Ralph* would have a winning strategy by generating an arbitrarily long sequence of $\mathcal{I}+\mathcal{I}'+\cdots+\mathcal{I}'$, and if the most complex subexpression of $\mathcal{J}$ was a shuffle, then $\mathrm{c}(\mathcal{J}) = (a', 0)$.) In this game *Ralph* can force *Felix* to match the derivations $\mathcal{I} \equiv \mathcal{I}+\mathcal{I}'$ and $\mathcal{J} \equiv \overleftarrow{\mathcal{J}'} + (\mathcal{J}' + \mathcal{J}'')$. Suppose that *Felix* does this with the respective derivations $\mathcal{J} \equiv (\overleftarrow{\mathcal{J}'}+\mathcal{K}_1)+(\mathcal{K}_2+m\mathcal{J}'+\mathcal{J}'')$ and $\mathcal{I} \equiv (\mathcal{I} + \mathcal{L}_1) + (\mathcal{L}_2 + k\mathcal{I}')$, where m and k are integers indicating repeated concatenation. As *Felix* is applying a winning strategy the following hold (where $A \sim B$ indicates *Felix* has a winning strategy in the game $\mathcal{G}(A, B)$):

$$\begin{array}{ll} \mathcal{I} \sim \overleftarrow{\mathcal{J}'} + \mathcal{K}_1 & \overleftarrow{\mathcal{J}'} \sim \mathcal{I} + \mathcal{L}_1 \\ \mathcal{I}' \sim \mathcal{K}_2 + m\mathcal{J}' + \mathcal{J}'' & \mathcal{J}' + \mathcal{J}'' \sim \mathcal{L}_2 + k\mathcal{I}' \\ \mathcal{I}' \sim \mathcal{L}_1 + \mathcal{L}_2 & \mathcal{J}' \sim \mathcal{K}_1 + \mathcal{K}_2 \end{array}$$

From these equations we can infer that $\mathcal{K}_1 \equiv \mathcal{J}''$ (applying the inductive hypothesis). The first two identities recursively reference each other, and by applying the induction hypothesis and the substitution principle we are able to derive

$$x\mathcal{I}' \equiv \mathcal{L}_1 + \mathcal{K}_1 \quad \text{and} \quad y\mathcal{J}' \equiv \mathcal{K}_1 + \mathcal{L}_1 \tag{6}$$

where x and y are integers indicating repeated concatenation. Then the equivalence $\mathcal{I} \equiv \mathcal{J}$ follows from the lead identities (Definition 4.8-9).

- $\mathcal{I} = \overrightarrow{\mathcal{I}'}$. This case is the mirror image of the previous case and may be treated in a similar way.
- $\mathcal{I} = \langle\{\mathcal{I}_1, \ldots, \mathcal{I}_m\}\rangle$. For any $\mathcal{J}$ such that *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$ we may assume that $\mathcal{J}$ is equivalent to some shuffle $\mathcal{J} \equiv \langle\{\mathcal{J}_1, \ldots, \mathcal{J}_n\}\rangle$, as $\mathrm{c}(\mathcal{J})$ must be 0 in its second element, and any subexpression that cannot be reduced by the shuffle identities (Definition 4.2-5) can easily be exploited by *Ralph* to give a winning strategy. *Ralph* can, for $i = 1 \ldots m$, play the derivation $\mathcal{I} \equiv \mathcal{I} + \mathcal{I}_i + \mathcal{I}$. The winning strategy of *Felix* must be able to respond with $\mathcal{J} \equiv \mathcal{J} + \mathcal{K}' + \mathcal{K} + \mathcal{K}'' + \mathcal{J}$ where *Felix* has a winning strategy in the game $\mathcal{G}(\mathcal{I}_i, \mathcal{K})$ and $\mathcal{K}' + \mathcal{K} + \mathcal{K}'' \equiv \mathcal{J}_j$ for some j. By the induction hypothesis we have $\mathcal{I}_i \equiv \mathcal{K}$, and furthermore, $\mathcal{I} \sqsubseteq \mathcal{K}', \mathcal{K}''$. A similar equivalence may be given for $\mathcal{J}_i$ for $i = 1 \ldots n$, so the result follows by the substitution rule and the first shuffle identity (Definition 4.5).

Therefore the induction holds and we are able to construct a full derivation of the equivalence between $\mathcal{I}$ and $\mathcal{J}$ from *Felix*'s winning strategy in the game $\mathcal{G}(\mathcal{I}, \mathcal{J})$. ■

THEOREM 1: Given any two model expressions $\mathcal{I}$ and $\mathcal{J}$ we have $\mathcal{I} \simeq \mathcal{J}$ if and only if $\mathcal{I} \equiv \mathcal{J}$.

Proof: The right to left direction ($\mathcal{I} \equiv \mathcal{J}$ implies $\mathcal{I} \simeq \mathcal{J}$) is simply Lemma 3. The opposite direction ($\mathcal{I} \simeq \mathcal{J}$ implies $\mathcal{I} \equiv \mathcal{J}$) corresponds to Lemma 4 which follows directly from Lemmas 6 and 9. Therefore the model expression algebra (Definition 4) is sound and complete. ■

VIII. Conclusion and Future Work

This work presents an important foundation in establishing a computational model for first order theories of linear order. We have seen that model expressions are expressively complete for first order theories of linear order [9], [3] and the algebra of Definition 4 presents a practical resource for reasoning directly about model expressions. A corollary of Theorem 1 is that determining equivalence of model-expressions is at most recursively enumerable. The question of whether this is a lower bound is left to future work.

References

[1] A. Pnueli, "The temporal logic of programs," in *Proceedings of the Eighteenth Symposium on Foundations of Computer Science*, 1977, pp. 46–57, providence, RI.

[2] T. French, J. C. McCabe-Dansted, and M. Reynolds, "Synthesis for temporal logic over the reals," in *Advances in Modal Logic*, 2012, pp. 217–238.

[3] H. Läuchli and J. Leonard, "On the elementary theory of linear order," *Fundamenta Mathematicae*, vol. 59, pp. 109–116, 1966.

[4] M. Reynolds, "Continuous temporal models," in *Australian Joint Conference on Artificial Intelligence*, ser. Lecture Notes in Computer Science, M. Stumptner, D. Corbett, and M. J. Brooks, Eds., vol. 2256. Springer, 2001, pp. 414–425.

[5] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3 – 34, 1995.

[6] D. Gabbay, I. Hodkinson, and M. Reynolds, *Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 1*. Oxford University Press, 1994.

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

[8] J. P. Burgess and Y. Gurevich, "The decision problem for linear temporal logic," *Notre Dame J. Formal Logic*, vol. 26, no. 2, pp. 115–128, 1985.

[9] T. French, J. McCabe-Dansted, and M. Reynolds, "Indiscrete models: Model building and model checking over linear time," in *Logic and Its Applications*, ser. Lecture Notes in Computer Science, K. Lodaya, Ed. Springer Berlin Heidelberg, 2013, vol. 7750, pp. 50–68.

[10] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, Massachusetts: The MIT Press, 1999.