Finite-state representations embodying temporal relations

Tim Fernando
Computer Science Department
Trinity College
Dublin 2, Ireland
`Tim.Fernando1@tcd.ie`

Abstract

Finite-state methods are applied to the Russell-Wiener-Kamp notion of time (based on events) and developed into an account of interval relations and semi-intervals. Strings are formed and collected in regular languages and regular relations that are argued to embody temporal relations in their various underspecified guises. The regular relations include retractions that reduce computations by projecting strings down to an appropriate level of granularity, and notions of partiality within and across such levels.

1 Introduction

It is a truism that to reason about change, some notion of time is useful to impose order on events. Less clear perhaps is whether or not time is shaped completely by the events it relates. An event-based notion of time going back to Russell and Wiener (Kamp and Reyle, 1993; Lück, 2006) is analyzed in the present work using finite-state methods that extend to interval relations, semi-intervals and granularity; e.g. (Allen, 1983; Freksa, 1992; Mani, 2007). Rather than take for granted some absolute (independent) notion of time (such as the real line), the basic approach is to form strings (from events and generalizations of events described below) and collect them in regular languages and regular relations. The claim is that this leads to a more satisfying account of the partiality of temporal information conveyed (for instance) in everyday speech. In particular, there is a sense (to be explained below) in which the strings, languages and relations of the approach embody a wide range of temporal relations that vary in degrees of underspecification. Those degrees depend on the events under consideration: the more events to relate, the finer grained time becomes.

Two temporal relations between events, called overlap $\bigcirc$ and (complete) precedence $\prec$, are employed in the Russell-Wiener-Kamp construction of time from events. To picture these relations between two events e and e', let us form the three "snapshots"

$$\boxed{e},\ \boxed{e'} \text{ and } \boxed{e,e'}$$

and arrange them much like a cartoon/film strip (with time progressing from left to right) to produce

$\boxed{e}\boxed{e'}$ as a record of e precedes e' (i.e., $e \prec e'$)
$\boxed{e'}\boxed{e}$ as a record of e' precedes e (i.e., $e' \prec e$)

and finally

$\boxed{e,e'}$ as a record of e overlaps e' (i.e., $e \bigcirc e'$) .

Formally, these strips are strings over the alphabet $Pow(\{e, e'\})$ of subsets of $\{e, e'\}$, with the curly braces in $\{e\}$, $\{e'\}$ and $\{e, e'\}$ redrawn as boxes to reinforce the construal of the subsets as snapshots. As explained in section 2 below, the three strings correspond exactly to the three (Russell-Wiener-Kamp) event structures over the events e and e', with a box in each string identifiable as a (Russell-Wiener-Kamp) temporal moment.[1]

[1]Briefly, $\bigcirc$ is just $\prec$-incomparability, and RWK-moments maximal antichains relative to $\prec$ (Lück, 2006). Details below, where we follow Kamp and Reyle (1993) in foregrounding $\bigcirc$.

Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing, pages 12–20,
Blois (France), July 12-15, 2011.

RWK	**Allen**	*Pow*$(\{e,e'\})^*$
$e \bigcirc e'$	$e \ \mathtt{=}\ e'$	$\boxed{e,e'}$
	$e \ \mathtt{s}\ e'$	$\boxed{e,e'}\boxed{e'}$
	$e \ \mathtt{si}\ e'$	$\boxed{e,e'}\boxed{e}$
	$e \ \mathtt{f}\ e'$	$\boxed{e'}\boxed{e,e'}$
	$e \ \mathtt{fi}\ e'$	$\boxed{e}\boxed{e,e'}$
	$e \ \mathtt{d}\ e'$	$\boxed{e'}\boxed{e,e'}\boxed{e'}$
	$e \ \mathtt{di}\ e'$	$\boxed{e}\boxed{e,e'}\boxed{e}$
	$e \ \mathtt{o}\ e'$	$\boxed{e}\boxed{e,e'}\boxed{e'}$
	$e \ \mathtt{oi}\ e'$	$\boxed{e'}\boxed{e,e'}\boxed{e}$
$e \prec e'$	$e \ \mathtt{m}\ e'$	$\boxed{e}\boxed{e'}$
	$e < e'$	$\boxed{e}\boxed{\ }\boxed{e'}$
$e' \prec e$	$e \ \mathtt{mi}\ e'$	$\boxed{e'}\boxed{e}$
	$e > e'$	$\boxed{e'}\boxed{\ }\boxed{e}$

Table 1: From Russell-Wiener-Kamp to Allen

But surely there are more relations than precedence $\prec$ and overlap $\bigcirc$ to consider — not to mention strings in *Pow*$(\{e,e'\})^*$ other than $\boxed{e}\boxed{e'}$, $\boxed{e'}\boxed{e}$ and $\boxed{e,e'}$. Inasmuch as event structures describe intervals, it is natural to ask about the thirteen different interval relations in Allen (1983). Evidently, there are nine ways for e and e' to overlap, and two ways (each) for e to precede e' (and e' to precede e). See Table 1, where strings are associated with Allen relations according to certain constructions presented below. Briefly, under these constructions, granularity can be refined by expanding the set of events related by $\prec$ and $\bigcirc$. In particular, it turns out that all thirteen Allen relations between e and e' fall out of the Russell-Wiener-Kamp construction (RWK) applied to an expansion of $\{e,e'\}$ by markers *pre*(e), *post*(e), *pre*(e'), *post*(e'), of the past and future of e and e', respectively. That is, RWK yields the Allen relations provided that, in McTaggart's terminology (McTaggart, 2008), we first enrich the B-series relations $\prec$ and $\bigcirc$ with A-series ingredients for tense. In the case of the Allen relation $e \ \mathtt{s}\ e'$, for instance, we get the string

$$\boxed{\mathit{pre}(e), \mathit{pre}(e')}\boxed{e,e'}\boxed{\mathit{post}(e), e'}\boxed{\mathit{post}(e), \mathit{post}(e')}$$

which a certain string function $\pi_{\{e,e'\}}$ maps to the Table 1 *Pow*$(\{e,e'\})^*$-entry

$$\boxed{e,e'}\boxed{e'}$$

for $e \ \mathtt{s}\ e'$. The rest of the Allen relations can be obtained similarly. The projection $\pi_{\{e,e\}}$ is one in a family of regular relations π_X that (as will be shown below) correspond to RWK under the aforementioned A-series enhancement. The subscript X indexing that family specifies the ingredients from which strings are formed, and (as a consequence) the granularity of temporal relations the strings embody. By varying that index X, we can overcome the limitations in any choice of finitely many events on which to construct event structures (i.e., $\bigcirc$ and $\prec$). What's more, for any set E (finite or infinite), we can represent every event structure over E in the inverse limit of the system of maps π_X, for X ranging over finite subsets of E.

More precisely, we start in section 2 with a careful presentation of event structures, extracting event structures $\mathbb{E}(\mathsf{s})$ from strings $\mathsf{s} \in$ *Pow*$(X)^*$ (over the alphabet of subsets of X) with A-series extensions $\mathsf{s}_\pm$ to capture the Allen relations in $\mathbb{E}(\mathsf{s}_\pm)$. A function on strings, block compression bc, is defined that gives canonical string representations $bc(\mathsf{s})$ of $\mathbb{E}(\mathsf{s}_\pm)$. In section 3, we transform bc into maps π_X, for different finite sets X of events, forming regular languages representing families of finite event structures, which are subsequently generalized and constrained.

Throughout what follows, strings are formed from subsets of some finite set X. An alternative considered in Karttunen (2005) is to flatten these subsets to strings, introducing brackets [and] to enclose temporal propositions understood to hold at the same period so that, for example, the string $\boxed{e,e'}\boxed{e'}$ of length 2 becomes the string [e e'] [e'] of length 7. It is easy to devise a finite-state transducer translating *Pow*$(X)^*$ to $(X \cup \{[,]\})^*$ in this way. A greater challenge is presented by brackets $[_a$ and $]_a$ decorated with granularities a (such as days or months or years) used in the analysis of calendar expressions in Niemi and Koskenniemi (2009). The approach below of structuring the symbols of the alphabet as sets simplifies many of the finite-state constructions of present interest.[2] An important example is *su-*

[2]As shown in section 3 below, the theme in Niemi and

perposition & (Fernando, 2004), a binary operation on strings over the alphabet *Pow*(X) that forms the componentwise union of strings of the same length

$$\alpha_1 \cdots \alpha_n \,\&\, \alpha'_1 \cdots \alpha'_n \stackrel{\text{def}}{=} (\alpha_1 \cup \alpha'_1) \cdots (\alpha_n \cup \alpha'_n)$$

(for $\alpha_i, \alpha'_i \subseteq X$). To illustrate,

$$\begin{array}{rcl} \boxed{e, e'} & = & \boxed{e} \,\&\, \boxed{e'} \\ \boxed{e}\boxed{e'} & = & \boxed{e}\boxed{} \,\&\, \boxed{}\boxed{e'} \\ \boxed{e'}\boxed{e} & = & \boxed{e'}\boxed{} \,\&\, \boxed{}\boxed{e} \,. \end{array}$$

A natural notion of containment between strings s and s' can be derived from & as follows. We say s *subsumes* s' and write $\mathsf{s} \trianglerighteq \mathsf{s}'$ if the strings have the same length, and the first is no different from its superposition with the second

$$\mathsf{s} \trianglerighteq \mathsf{s}' \stackrel{\text{def}}{\iff} \mathsf{s} \text{ and } \mathsf{s}' \text{ have the same length, and } \mathsf{s} = \mathsf{s} \,\&\, \mathsf{s}' \,.$$

That is, $\trianglerighteq$ is componentwise inclusion $\supseteq$ between strings of the same length,

$$\alpha_1 \cdots \alpha_n \trianglerighteq \alpha'_1 \cdots \alpha'_n \iff \alpha_i \supseteq \alpha'_i \text{ for } 1 \leq i \leq n \,.$$

To compare strings of different lengths, we *unpad*, stripping off initial and final empty boxes $\Box$

$$\mathit{unpad}(\mathsf{s}) \stackrel{\text{def}}{=} \begin{cases} \mathit{unpad}(\mathsf{s}') & \text{if } \mathsf{s} = \Box\mathsf{s}' \text{ or else if } \mathsf{s} = \mathsf{s}'\Box \\ \mathsf{s} & \text{otherwise} \end{cases}$$

so that, for example,

$$\mathit{unpad}(\boxed{}^{n}\boxed{e}\boxed{}\boxed{e, e'}\boxed{}^{m}) = \boxed{e}\boxed{}\boxed{e, e'}$$

for all integers $n, m \geq 0$. Now, using the equivalence $\approx$ between strings that *unpad* maps alike

$$\mathsf{s} \approx \mathsf{s}' \stackrel{\text{def}}{\iff} \mathit{unpad}(\mathsf{s}) = \mathit{unpad}(\mathsf{s}') \,,$$

we generalize subsumption $\trianglerighteq$ to *containment* $\sqsupseteq$, taking $\mathsf{s} \sqsupseteq \mathsf{s}'$ (read: s *contains* s') to mean that s subsumes some string *unpad*-equivalent to s'

$$\mathsf{s} \sqsupseteq \mathsf{s}' \stackrel{\text{def}}{\iff} (\exists \mathsf{s}'' \approx \mathsf{s}')\, \mathsf{s} \trianglerighteq \mathsf{s}'' \,.$$

Koskenniemi (2009) of composing finite-state transducers can be developed with symbols structured as sets, and regular relations as retractions.

(A_1)	$e \bigcirc e$ (i.e. $\bigcirc$ is reflexive)
(A_2)	$e \bigcirc e' \implies e' \bigcirc e$
(A_3)	$e \prec e' \implies$ not $e \bigcirc e'$
(A_4)	$e \prec e'$ and $e' \bigcirc e''$ and $e'' \prec e''' \implies e \prec e'''$
(A_5)	$e \prec e'$ or $e \bigcirc e'$ or $e' \prec e$

Table 2: Axioms for (RWK) event structures

Thus, if s contains s' (e.g. if s is s'), then so do *unpad*(s) and $\mathsf{s}''\mathsf{s}$ and $\mathsf{s}\mathsf{s}''$ (for all s''). It will be convenient to extend $\sqsupseteq$ to languages L, conceived as disjunctions, agreeing that s *contains* L if s contains some element of L

$$\mathsf{s} \sqsupseteq L \stackrel{\text{def}}{\iff} (\exists \mathsf{s}' \in L)\, \mathsf{s} \sqsupseteq \mathsf{s}'$$

so that $\mathsf{s} \sqsupseteq \mathsf{s}'$ iff $\mathsf{s} \sqsupseteq \{\mathsf{s}'\}$. Containment $\sqsupseteq$ is applied to event structures in section 2, with different sets X of events related by projections π_X in section 3. Containment is also useful when sidestepping completeness assumptions built into event structures and π_X, as we shall see.

2 Event structures from strings

A *(Russell-Wiener-Kamp) event structure* (Kamp and Reyle, 1993) is a triple $\langle E, \bigcirc, \prec \rangle$ consisting of a set E of events, and two binary relations on E, *(temporal) overlap* $\bigcirc$ and *(complete) precedence* $\prec$ satisfying axioms (A_1) to (A_5) in Table 2. To get a sense for what these axioms mean, it is useful to interpret them relative to triples $\langle E^{\mathsf{s}}, \bigcirc^{\mathsf{s}}, \prec^{\mathsf{s}} \rangle$ defined from strings s of sets as follows. We put into E^{s} each e that occurs in s

$$E^{\mathsf{s}} \stackrel{\text{def}}{=} \{e \mid \mathsf{s} \sqsupseteq \boxed{e}\}$$

and define e to s-*overlap* e' precisely if e and e' share a box in s

$$e \bigcirc^{\mathsf{s}} e' \stackrel{\text{def}}{\iff} \mathsf{s} \sqsupseteq \boxed{e, e'} \,.$$

As $\boxed{e, e} = \boxed{e}$ and $\boxed{e, e'} = \boxed{e', e}$, it follows that ($A_1$) and ($A_2$) are true for $\bigcirc = \bigcirc^{\mathsf{s}}$. Next, we say e s-*precedes* e' if e occurs in s to the left of e' but never in the same box as e' or to the right of e'

$$e \prec^{\mathsf{s}} e' \stackrel{\text{def}}{\iff} \mathsf{s} \sqsupseteq \boxed{e}\boxed{}^{*}\boxed{e'} \text{ and not } \mathsf{s} \sqsupseteq \boxed{e, e'} \mid \boxed{e'}\boxed{}^{*}\boxed{e}$$

(where | is non-deterministic choice, often written $+$). It is easy to see that together $\bigcirc^{\mathsf{s}}$ and $\prec^{\mathsf{s}}$ validate (A_3) and (A_4). This leaves (A_5), a counter-example to which is provided by the string $\boxed{e}\boxed{e'}\boxed{e}$. With this in mind, we define an element $e \in E^{\mathsf{s}}$ to be an s-*interval* if for $\mathsf{s} = \alpha_1 \cdots \alpha_n$,

$$e \in \alpha_i \cap \alpha_j \text{ and } i \leq k \leq j \quad \Longrightarrow \quad e \in \alpha_k$$

for all integers i, j, k from 1 to n. We call s *structural* if every $e \in E^{\mathsf{s}}$ is an s-interval. If s is structural, then

$$e \prec^{\mathsf{s}} e' \quad \Longleftrightarrow \quad \mathsf{s} \sqsupseteq \boxed{e}\boxed{}^{*}\boxed{e'} \text{ and not } \mathsf{s} \sqsupseteq \boxed{e, e'}\,.$$

Moreover, we have

Proposition 1. *If* s *is structural, then* $\langle E^{\mathsf{s}}, \bigcirc^{\mathsf{s}}, \prec^{\mathsf{s}} \rangle$ *is an event structure.*

As a string s need not be structural, it is useful to define the subset $I(\mathsf{s})$ of E^{s} consisting of s-intervals

$$I(\mathsf{s}) \quad \stackrel{\text{def}}{=} \quad \{e \in E^{\mathsf{s}} \mid e \text{ is an } \mathsf{s}\text{-interval}\}\ .$$

For example,

$$I(\boxed{e}\boxed{e'}\boxed{e}) \quad = \quad \{e'\}\ .$$

Next, for any set X, we define the function ρ_X on strings (of sets) to componentwise intersect with X

$$\rho_X(\alpha_1 \cdots \alpha_n) \quad \stackrel{\text{def}}{=} \quad (\alpha_1 \cap X) \cdots (\alpha_n \cap X)$$

so that, for instance, if $\hat{\mathsf{s}}$ is $\boxed{e}\boxed{e'}\boxed{e}$,

$$\rho_{I(\hat{\mathsf{s}})}(\hat{\mathsf{s}}) \quad = \quad \boxed{}\boxed{e'}\boxed{}\ .$$

In general, $\rho_{I(\mathsf{s})}(\mathsf{s})$ is structural for all strings s. Setting $i(\mathsf{s})$ to $\rho_{I(\mathsf{s})}(\mathsf{s})$, and $\mathbb{E}(\mathsf{s})$ to the triple $\langle E^{i(\mathsf{s})}, \bigcirc^{i(\mathsf{s})}, \prec^{i(\mathsf{s})} \rangle$ induced by $i(\mathsf{s})$, we note

Corollary 2. $\mathbb{E}(\mathsf{s})$ *is an event structure for every string* s *of sets.*

An obvious question Corollary 2 raises is: can every event structure over a set E be presented as $\mathbb{E}(\mathsf{s})$ for a suitable string $\mathsf{s} \in \mathit{Pow}(E)^*$? For infinite sets E, more methods are clearly needed — and considered in the next section. As for finite E, an affirmative answer follows from Russell-Wiener-Kamp (RWK, Kamp and Reyle, 1993), which we now briefly recall. Given an event structure $\langle E, \bigcirc, \prec \rangle$, we construct a linear order $\langle T_{\bigcirc}, \prec_T \rangle$ as follows. The set $T_{\bigcirc}$ of *(RWK) temporal moments* consists of subsets t of E that pairwise $\bigcirc$-overlap

$$(\forall e, e' \in t) \qquad e \bigcirc e'$$

and are $\subseteq$-maximal among such subsets

$$(\forall e \in E) \qquad \text{if } (\forall e' \in t)\ e \bigcirc e' \text{ then } e \in t\ .$$

For $t, t' \in T_{\bigcirc}$, we then put $t \prec_T t'$ if some element of t $\prec$-precedes some element of t'

$$t \prec_T t' \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad (\exists e \in t)(\exists e' \in t')\ e \prec e'\ .$$

One can then show that not only does $\prec_T$ linearly order $T_{\bigcirc}$, but that relative to that linear order, every $e \in E$ defines an interval

$$e \in t \qquad \text{whenever } e \in t_1 \text{ and } e \in t_2 \\ \text{for some } t_1, t_2 \text{ with } t_1 \prec_T t \prec_T t_2$$

and the relations $\bigcirc$ and $\prec$ can be interpreted as overlap

$$e \bigcirc e' \quad \Longleftrightarrow \quad (\exists t \in T_{\bigcirc})\ e \in t \text{ and } e' \in t$$

and complete precedence

$$e \prec e' \quad \Longleftrightarrow \quad (\forall t \in T_{\bigcirc})(\forall t' \in T_{\bigcirc}) \\ e \in t \text{ and } e' \in t' \text{ implies } t \prec_T t'\ .$$

To illustrate, the three event structures on $E = \{e, e'\}$ yield three linear orders $\langle T_{\bigcirc}, \prec_T \rangle$ that can be pictured as the three strings $\boxed{e, e'}$, $\boxed{e}\boxed{e'}$ and $\boxed{e'}\boxed{e}$.

But then what about the ten other strings in Table 1 and the various Allen relations? Each of these strings violates the $\subseteq$-maximality requirement on $T_{\bigcirc}$ above. We can neutralize that requirement by adjoining pre- and post-events, turning, for instance,

$$\boxed{e}\boxed{e, e'}\boxed{e'} \quad \text{into} \quad \boxed{e, \mathit{pre}(e')}\boxed{e, e'}\boxed{e', \mathit{post}(e)}\ .$$

On structural strings, $\mathit{pre}(e)$ and $\mathit{post}(e)$ negate e, whilst preserving structurality. More precisely, given a set E, let

$$E_{\pm} \quad \stackrel{\text{def}}{=} \quad E \ \cup\ \{\mathit{pre}(e) \mid e \in E\} \\ \cup\ \{\mathit{post}(e) \mid e \in E\}$$

and call a string $\mathsf{s} = \alpha_1\alpha_2\cdots\alpha_n$ *E-delimited* if for all $e \in E$ and $i \in \{1, 2, \ldots, n\}$,

$$
\begin{aligned}
\mathit{pre}(e) \in \alpha_i \quad &\iff \quad \mathsf{s} \sqsupseteq \boxed{e} \text{ but } \alpha_1\cdots\alpha_i \not\sqsupseteq \boxed{e} \\
&(\iff \quad e \in (\bigcup_{j=i+1}^{n} \alpha_j) - \bigcup_{j=1}^{i} \alpha_j)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathit{post}(e) \in \alpha_i \quad &\iff \quad \mathsf{s} \sqsupseteq \boxed{e} \text{ but } \alpha_i\cdots\alpha_n \not\sqsupseteq \boxed{e} \\
&(\iff \quad e \in (\bigcup_{j=1}^{i-1} \alpha_j) - \bigcup_{j=i}^{n} \alpha_j) \ .
\end{aligned}
$$

It is immediate that for every string $\mathsf{s} \in \mathit{Pow}(E)^*$, there is a unique E-delimited string $\mathsf{s}' \in \mathit{Pow}(E_\pm)^*$ such that $\rho_E(\mathsf{s}') = \mathsf{s}$. Let $\mathsf{s}_\pm$ be that unique string.

Proposition 3. *For every finite set E, there is a finite-state transducer that computes the map $\mathsf{s} \mapsto \mathsf{s}_\pm$ from $\mathit{Pow}(E)^*$ to $\mathit{Pow}(E_\pm)^*$.*

If s is structural, then so is $\mathsf{s}_\pm$ — making $\langle E^{\mathsf{s}_\pm}, \bigcirc^{\mathsf{s}_\pm}, \prec^{\mathsf{s}_\pm} \rangle$ an event structure (for structural s). Extending a string $\mathsf{s} \in \mathit{Pow}(\{e, e'\})^*$ to $\mathsf{s}_\pm$ leads to a refinement of $\bigcirc$ and $\prec$ to any of the 13 Allen relations — e.g. whenever e and e' are s-intervals,

$$
\begin{aligned}
e \ \mathsf{d}^{\mathsf{s}} \ e' \quad &\iff \quad \mathit{pre}(e) \bigcirc^{\mathsf{s}_\pm} e' \text{ and } e \bigcirc^{\mathsf{s}_\pm} e' \\
&\qquad\quad \text{and } \mathit{post}(e) \bigcirc^{\mathsf{s}_\pm} e' \\
e <^{\mathsf{s}} e' \quad &\iff \quad e \prec^{\mathsf{s}_\pm} e' \text{ and} \\
&\qquad\quad \mathit{post}(e) \bigcirc^{\mathsf{s}_\pm} \mathit{pre}(e') \ .
\end{aligned}
$$

Given that there is a finite-state transducer for the map $\mathsf{s} \mapsto \mathsf{s}_\pm$, it is tempting to leave out the pre- and post-events for simplicity.

The map $\mathsf{s} \mapsto \mathsf{s}_\pm$ aside, different strings $\mathsf{s} \in \mathit{Pow}(E)^*$ can give the same event structure $\mathbb{E}(\mathsf{s})$. Take, for example, the strings in $\boxed{e}^+\boxed{e'}^+$ (where $L^+ \overset{\text{def}}{=} L^*L$), each of which gives the event structure pictured by $\boxed{e}\boxed{e'}$. In general, let us reduce all adjacent identical boxes $\alpha\alpha^n$ to one α in the *block compression* $\mathit{bc}(\mathsf{s})$ *of* a string s

$$
\mathit{bc}(\mathsf{s}) \quad \overset{\text{def}}{=} \quad \begin{cases} \mathit{bc}(\alpha\mathsf{s}') & \text{if } \mathsf{s} = \alpha\alpha\mathsf{s}' \\ \alpha\,\mathit{bc}(\alpha'\mathsf{s}') & \text{if } \mathsf{s} = \alpha\alpha'\mathsf{s}' \text{ with } \alpha \neq \alpha' \\ \mathsf{s} & \text{otherwise} \end{cases}
$$

so that, for example,

$$
\mathit{bc}(\mathsf{s}) = \boxed{e}\boxed{e'} \qquad \text{for every } \mathsf{s} \in \boxed{e}^+\boxed{e'}^+ \ .
$$

The map bc is a regular relation, and implements the slogan "no time without change" (Kamp and Reyle 1993, page 674). Clearly, bc does *not* alter the event structure $\mathbb{E}(\mathsf{s})$ represented by a string s

$$
\mathbb{E}(\mathit{bc}(\mathsf{s})) \quad = \quad \mathbb{E}(\mathsf{s}) \ .
$$

Neither does unpadding, which suggests defining a function π that unpads after (or equivalently: before) block compression

$$
\pi(\mathsf{s}) \quad \overset{\text{def}}{=} \quad \mathit{unpad}(\mathit{bc}(\mathsf{s})) \quad [= \mathit{bc}(\mathit{unpad}(\mathsf{s}))]
$$

so that, for example,

$$
\pi(\mathsf{s}) = \boxed{e}\boxed{e'} \qquad \text{for every } \mathsf{s} \in \boxed{}^*\boxed{e}^+\boxed{e'}^+\boxed{}^* \ .
$$

Before using π to define the functions π_X in the next section, let us note that on delimited strings $\mathsf{s}_\pm$, π captures what is essential for representing event structures.

Proposition 4. For structural strings s and $\mathsf{s}' \in \mathit{Pow}(E)^*$, the following four conditions, (a) to (d), are equivalent

(a) $\mathit{bc}(\mathsf{s}) = \mathit{bc}(\mathsf{s}')$

(b) $\mathbb{E}(\mathsf{s}_\pm) = \mathbb{E}(\mathsf{s}'_\pm)$

(c) $\mathit{bc}(\mathsf{s}_\pm) = \mathit{bc}(\mathsf{s}'_\pm)$

(d) $\pi(\mathsf{s}_\pm) = \pi(\mathsf{s}'_\pm)$.

It follows from Proposition 4 that for structural $\mathsf{s} \in \mathit{Pow}(E)^*$,

$$
\mathbb{E}(\mathsf{s}_\pm) \ = \ \mathbb{E}(\mathit{bc}(\mathsf{s}_\pm)) \ = \ \mathbb{E}(\pi(\mathsf{s}_\pm))
$$

as $\mathit{bc}(\mathit{bc}(\mathsf{s}_\pm)) = \mathit{bc}(\mathsf{s}_\pm) = \pi(\mathsf{s}_\pm)$.

3 Varying X with retractions π_X and generalizations

Fix some large set E, and let $\Sigma = \mathit{Pow}(E)$ be the alphabet from which we form strings. Given a language $L \subseteq \Sigma^*$ and a function $f : \Sigma^* \to \Sigma^*$ on strings over Σ, we write $f[L]$ for the f-image of L

$$
f[L] \quad \overset{\text{def}}{=} \quad \{f(\mathsf{s}) \mid \mathsf{s} \in L\}
$$

and $f^{-1}L$ for the inverse f-image of L

$$
f^{-1}L \quad \overset{\text{def}}{=} \quad \{\mathsf{s} \in \Sigma^* \mid f(\mathsf{s}) \in L\} \ .
$$

Applying these constructions in sequence, note that

$$f^{-1}f[L] \quad = \quad \{\mathsf{s} \in \Sigma^* \mid (\exists \mathsf{s}' \in L)\ f(\mathsf{s}) = f(\mathsf{s}')\}$$

which we shall refer to as the *f-closure of L*, L^f

$$L^f \quad \stackrel{\text{def}}{=} \quad f^{-1}f[L]$$

(as $L \subseteq L^f = L^{ff}$ and the operation preserves $\subseteq$-inclusion: $L \subseteq L'$ implies $L^f \subseteq L'^f$). In this section, we form f-closures for different f's computed by finite-state transducers (assuming E is finite), including the functions *unpad*, bc, π, and ρ_X (for subsets X of E; recall $\rho_X(\alpha_1 \cdots \alpha_n) = (\alpha_1 \cap X) \cdots (\alpha_n \cap X)$). Putting these together, let $\pi_X : \textit{Pow}(E)^* \to \textit{Pow}(X)^*$ be the composition $\rho_X ; \pi$ of ρ_X followed by π

$$\pi_X(\mathsf{s}) \quad \stackrel{\text{def}}{=} \quad \pi(\rho_X(\mathsf{s})) \quad = \quad \textit{unpad}(bc(\rho_X(\mathsf{s})))$$

so that for every $\mathsf{s} \in \textit{Pow}(E)^*$ and $e \in E$, e is an s-interval iff $\pi_{\{e\}}(\mathsf{s}) = \boxed{e}$.

To study an event alongside other events, we generalize the superposition operation & (defined in the introduction) from strings of the same length to languages over the alphabet Σ. First, we collect superpositions $\mathsf{s} \,\&\, \mathsf{s}'$ of strings s and s' of the same length from languages L and L' in the *superposition*

$$L \,\&\, L' \quad \stackrel{\text{def}}{=} \quad \bigcup_{n \geq 0} \{\mathsf{s} \,\&\, \mathsf{s}' \mid \mathsf{s} \in L \cap \Sigma^n \text{ and } \mathsf{s}' \in L' \cap \Sigma^n\}$$

(Fernando, 2004). We then form the superposition of the f-closures of L and L', and take its f-image for the *f-superposition* $L \,\&_f\, L'$

$$L \,\&_f\, L' \quad \stackrel{\text{def}}{=} \quad f[L^f \,\&\, L'^f]\ .$$

For example, the π-superposition $\boxed{e} \,\&_\pi\, \boxed{e'}$ consists of the 13 strings in Table 1, which can be divided up as follows. Put the 9 ways for e and e' to overlap (according to Allen) in

$$\begin{aligned}\mathcal{A}(e \bigcirc e') \quad &\stackrel{\text{def}}{=} \quad (\epsilon \mid \boxed{e} \mid \boxed{e'})\boxed{e, e'}(\epsilon \mid \boxed{e} \mid \boxed{e'}) \\ &= \quad \boxed{e, e'} \mid \boxed{e, e'}\boxed{e} \mid \boxed{e, e'}\boxed{e'} \mid \cdots \\ &\qquad \mid \boxed{e'}\boxed{e, e'}\boxed{e'}\end{aligned}$$

(where ϵ is the empty string), and the 2 ways for e to precede e' in

$$\mathcal{A}(e \prec e') \quad \stackrel{\text{def}}{=} \quad \boxed{e}\boxed{e'} \mid \boxed{e}\boxed{}\boxed{e'}\ .$$

All 13 strings then end up in

$$\begin{aligned}\boxed{e} \,\&_\pi\, \boxed{e'} \quad = \quad &\mathcal{A}(e \prec e') \mid \mathcal{A}(e \bigcirc e') \mid \\ &\mathcal{A}(e' \prec e)\end{aligned}$$

in accordance with axiom (A_5) in Table 2. Stepping from two to any finite number $n \geq 1$ of events $e_1, \ldots, e_n$ in E (where $\Sigma = \textit{Pow}(E)$), let us define languages $\mathcal{E}(e_1 \cdots e_n)$ by induction on n as follows

$$\begin{aligned}\mathcal{E}(e_1) \quad &\stackrel{\text{def}}{=} \quad \boxed{e_1} \\ \mathcal{E}(e_1 \cdots e_{n+1}) \quad &\stackrel{\text{def}}{=} \quad \mathcal{E}(e_1 \cdots e_n) \,\&_\pi\, \boxed{e_{n+1}}\end{aligned}$$

(for $n \geq 1$). Recalling that $I(\mathsf{s})$ denotes the set of s-intervals, we can generalize the equation

$$I(\mathsf{s}) \quad = \quad \{e \in E^{\mathsf{s}} \mid \pi_{\{e\}}(\mathsf{s}) = \boxed{e}\}$$

as follows.

Proposition 5. *For every $\mathsf{s} \in \textit{Pow}(E)^*$ and every finite subset $\{e_1, \ldots, e_n\}$ of E, all e_i's are s-intervals iff $\pi_{\{e_1,\ldots,e_n\}}$ maps s to a string in $\mathcal{E}(e_1 \cdots e_n)$*

$$\{e_1, \ldots, e_n\} \subseteq I(\mathsf{s}) \iff \pi_{\{e_1,\ldots,e_n\}}(\mathsf{s}) \in \mathcal{E}(e_1 \cdots e_n)\ .$$

We can bring out the f-closures behind Proposition 5 by defining a language $L \subseteq \Sigma^*$ to be *f-closed* if its f-closure L^f is a subset of L. As it is always the case that $L \subseteq f^{-1}f[L]$,

$$L \text{ is } f\text{-closed} \iff L^f = L\ .$$

According to Proposition 5, the set $\mathcal{I}(e_1 \cdots e_n)$ of strings s such that each e_i is an s-interval (for $1 \leq i \leq n$) is $\pi_{\{e_1,\ldots,e_n\}}$-closed, and what's more, its $\pi_{\{e_1,\ldots,e_n\}}$-image is a subset of (in fact, identical to) $\mathcal{E}(e_1 \cdots e_n)$. Observe that a language L is f-closed iff for all $\mathsf{s} \in \Sigma^*$,

$$\mathsf{s} \in L \iff f(\mathsf{s}) \in f[L]$$

which constitutes a reduction in the cost of checking membership in L insofar as the f-image $f[L]$ of

L is a reduction of L (and the computational cost of f can be ignored). In the case of Proposition 5, whereas $\mathcal{I}(e_1 \cdots e_n)$ is infinite, $\mathcal{E}(e_1 \cdots e_n)$ is finite. Focusing on the case $n = 2$, note that the relations of overlap $\bigcirc$ and precedence $\prec$ between e and e' are $\pi_{\{e,e'\}}$-closed in that

Proposition 6. *For every* $\mathsf{s} \in \mathit{Pow}(E)^*$ *such that* $e, e' \in E$ *are* s*-intervals,*

$$e \bigcirc^{\mathsf{s}} e' \iff \pi_{\{e,e'\}}(\mathsf{s}) \in \mathcal{A}(e \bigcirc e')$$
$$e \prec^{\mathsf{s}} e' \iff \pi_{\{e,e'\}}(\mathsf{s}) \in \mathcal{A}(e \prec e') .$$

Under the appropriate definitions, the 13 Allen relations between e and e' are also $\pi_{\{e,e'\}}$-closed. A notion for which π_X-closedness is problematic, however, is the following. We say $e \in E$ is *left-bounded in* s if s is a non-empty string such that $e \not\in \alpha$ where α is the first symbol of s — or equivalently, $\mathit{pre}(e) \in E^{\mathsf{s}\pm}$. Although the set of strings s in which e is left-bounded is not $\pi_{\{e\}}$-closed, the equivalences

$$e \not\in \alpha \iff \mathit{bc}(\rho_{\{e\}}(\mathsf{s})) \in (\boxed{\ }\boxed{e})^+(\boxed{\ }|\epsilon)$$

(where α is the first symbol of the non-empty string s) and

$$\mathit{pre}(e) \in E^{\mathsf{s}\pm} \iff \pi_{\{\mathit{pre}(e)\}}(\mathsf{s}_\pm) = \boxed{\mathit{pre}(e)}$$

give two different functions f for which the set is f-closed — viz., the composition $\rho_{\{e\}}; \mathit{bc}$ of $\rho_{\{e\}}$ followed by bc, and the composition $\cdot_\pm; \pi_{\{\mathit{pre}(e)\}}$ of the map $\mathsf{s} \mapsto \mathsf{s}_\pm$ followed by $\pi_{\{\mathit{pre}(e)\}} = \rho_{\{\mathit{pre}(e)\}}; \mathit{bc}; \mathit{unpad}$. Note

Proposition 7. *If* $f = g; h$ *where* $g; g = g$ *then every* f*-closed language is* g*-closed.*

The cascade of regular relations above is reminiscent of Niemi and Koskenniemi (2009), with each successive function reducing the input. The case of left-boundedness suggests caution against over-reducing; the map *unpad* (separating bc from π) abstracts away temporal span. To see that $\&_{\mathit{bc}}$ gives us more control than $\&_\pi$, let us reformulate the example (from (Niemi and Koskenniemi, 2009)) of the 12 months of year 2008 in our framework as

$$\boxed{\text{y2008}} \;\&_{\mathit{bc}}\; \boxed{\text{Jan}}\boxed{\text{Feb}}\boxed{\text{Mar}}\cdots\boxed{\text{Dec}}$$
$$= \boxed{\text{y2008,Jan}}\boxed{\text{y2008,Feb}}\boxed{\text{y2008,Mar}}\cdots\boxed{\text{y2008,Dec}}$$

which $\rho_{\{\text{y2008}\}}; \mathit{bc}$ maps back to $\boxed{\text{y2008}}$. Given a function f such that $f; f = f$ and a subset X of E, we may call the composition $f_X \stackrel{\text{def}}{=} \rho_X; f$ of ρ_X with f a *retraction* insofar as f_X preserves the structure $\&_f$ introduces

$$f_X(\mathsf{s} \,\&_f\, \mathsf{s}') \;=\; f_X(\mathsf{s}) \,\&_f\, f_X(\mathsf{s}')$$

(where a string s is, as usual, conflated with the language $\{\mathsf{s}\}$). Let us say a language L is X-*determined* if L is ρ_X-closed. By Proposition 7, f_X-closed languages are X-determined. Moreover, the totality of finite subsets X of E (partially ordered by $\subseteq$) indexes an inverse system of maps π_X, the inverse limit of which represents every event structure over E. This fact bolsters the claim of embodiment made in the title of the present paper, reinforcing (as it does) the notion that strings are full-blooded semantic entities (familiar already from Linear Temporal Logic, where they can be viewed as Kripke models.; e.g. (Emerson, 1990)). Is the choice of π in the inverse system π_X sacrosanct? Should we not perhaps stop short of π_X at bc_X to capture, for instance, left-bounded events e? Not necessarily. Delimiting $\mathsf{s} \mapsto \mathsf{s}_\pm$ before applying $\pi_{\{e,\mathit{pre}(e)\}}$, we have

$$e \text{ is left-bounded in } \mathsf{s} \iff \pi_{\{e,\mathit{pre}(e)\}}(\mathsf{s}_\pm) \in \boxed{\mathit{pre}(e)}(\boxed{e}\boxed{\ })^*\boxed{e} .$$

But should we take it for granted that s amounts to $\mathsf{s}_\pm$?

Not if a string s is to embody underspecification, so that s may represent, relative to some background set C of strings, the set $C[\mathsf{s}]$ of strings in C that $\sqsupseteq$-contain s

$$C[\mathsf{s}] \stackrel{\text{def}}{=} \{\mathsf{s}' \in C \mid \mathsf{s}' \sqsupseteq \mathsf{s}\}$$

(a regular language, provided C is). Recall, for instance, the interest in representing cognitively natural disjunctions of Allen relations (Freksa, 1992). Under the present framework, some such disjunctions can be read off strings. For example, overlap between e and e' described by the (disembodied) abstract expression "$e \bigcirc e'$" is embodied by the box $\boxed{e, e'}$. That is, the set of strings s such that $e \bigcirc^{\mathsf{s}} e'$ is $C[\mathsf{s}]$ for $C = \Sigma^*$ and $\mathsf{s} = \boxed{e, e'}$. What about the precedence $e \prec e'$? This is where $\mathit{pre}(e')$ and *post*(e) are helpful. Form $C[\mathsf{s}]$ where s is the (non-delimited) string

$$\boxed{e, pre(e')}\boxed{post(e)}$$

and C is the language $\{\mathsf{s}'_{\pm} \mid \mathsf{s}' \in Pow(E)^*\}$. This language C and many more constraints can be formulated in finite-state terms familiar from say, Beesley and Karttunen (1983), as shown in Fernando (2011). Auxiliary constructs such as *pre*(e) and *post*(e) (that may later be dropped) have proved enormously useful tools advancing finite-state methods. Rather than claim for these constructs the same ontological status that events in E may enjoy, however, we might reconstrue the elements in boxes not as concrete particulars but rather as temporal propositions with possibly scattered occurrences (instead of the restriction to intervals characteristic of event structures). This would allow us to introduce a negation of e without requiring that the temporal projection of e or its complement be an interval. (Moreover, recalling the calendar example of Jan, Feb, ..., Dec above, we may well want to form a string s such that none of Jan, Feb, ..., Dec are s-intervals.)

That said, the families $\mathcal{E}(e_1 \cdots e_n)$ of regular languages above extend to

$$\mathcal{S}(e) \stackrel{\text{def}}{=} \boxed{pre(e)}(\epsilon \mid \boxed{})\boxed{e}(\epsilon \mid \boxed{})\boxed{post(e)}$$

and for $n \geq 1$,

$$\mathcal{S}(e_1 \cdots e_{n+1}) \stackrel{\text{def}}{=} \mathcal{S}(e_1 \cdots e_n) \,\&_\pi\, \mathcal{S}(e_{n+1})$$

with uncertainty injected $\boxed{}$ at the semi-intervals $\boxed{pre(e)}\boxed{e}$ and $\boxed{e}\boxed{post(e)}$, so that strings in $\mathcal{S}(e_1 e_2)$ embody disjunctions of Allen relations between e_1 and e_2. For example, we can represent temporal inclusion of e_1 within e_2 by the string

$$\boxed{pre(e_1), pre(e_2)}\boxed{pre(e_1)}\boxed{e_1, e_2}\boxed{post(e_1)}\boxed{post(e_1), post(e_2)}$$

(of length 5) in $\mathcal{S}(e_1 e_2)$, instead of the four strings from $\mathcal{E}(e_1 e_2)$ for $e_1 \; \mathtt{R} \; e_2$, $\mathtt{R} \in \{\mathtt{=}, \mathtt{s}, \mathtt{f}, \mathtt{d}\}$. Resisting the step from s to $\mathsf{s}_{\pm}$ leaves room for a form of underspecification that is natural for strings *qua* extensions (denotations), if not indices (Fernando, 2011).

4 Conclusion

The sense of embodiment claimed by the present paper boils down to reducing chronological order to succession within a string of boxes. But what boxes? That depends on our interests. Were we interested in months, then we might portray a year as the string

$$\mathsf{s}_{yr/mo} \stackrel{\text{def}}{=} \boxed{\mathsf{Jan}}\boxed{\mathsf{Feb}}\boxed{\mathsf{Mar}}\cdots\boxed{\mathsf{Dec}}$$

of length 12. Or were we also interested in days, perhaps the string

$$\mathsf{s}_{yr/mo,dy} \stackrel{\text{def}}{=} \boxed{\mathsf{Jan,d1}}\boxed{\mathsf{Jan,d2}}\cdots\boxed{\mathsf{Dec,d31}}$$

of length 365 (for a non-leap year). Observe that for $X = \{\mathsf{Jan,Feb,\ldots,Dec}\}$,

$$\pi_X(\mathsf{s}_{yr/mo,dy}) = \mathsf{s}_{yr/mo}$$

and that we can picture the $\mathsf{s}_{yr/mo,dy}$-intervalhood of Jan by the equation

$$\pi_{\{\mathsf{Jan}\}}(\mathsf{s}_{yr/mo,dy}) = \boxed{\mathsf{Jan}}$$

in contrast to d1, for which

$$\pi_{\{\mathsf{d1}\}}(\mathsf{s}_{yr/mo,dy}) = (\boxed{\mathsf{d1}}\boxed{})^{11}\boxed{\mathsf{d1}} .$$

In general, e is an s-interval precisely if $\pi_{\{e\}}$ maps s to $\boxed{e}$

$$e \in I(\mathsf{s}) \iff \pi_{\{e\}}(\mathsf{s}) = \boxed{e} .$$

Hence, all of $e_1, e_2, \ldots, e_n$ are s-intervals if $\mathsf{s} \in \mathcal{E}_\circ(e_1 \cdots e_n)$ where

$$\mathcal{E}_\circ(e_1 \cdots e_n) \stackrel{\text{def}}{=} \bigcap_{i=1}^{n} \pi^{-1}_{\{e_i\}}\boxed{e_i} .$$

That is, we can form the regular languages

$$\mathcal{E}(e_1 \cdots e_n) = \pi_{\{e_1,\ldots,e_n\}}[\mathcal{E}_\circ(e_1 \cdots e_n)]$$

starring in Proposition 5, without ever mentioning $\&$ or $\trianglerighteq$. More importantly, the regular relations π_X apply with or without the constraint of intervalhood imposed by RWK event structures. Furthermore, a bounded level of granularity is supported that we can adjust through X, as illustrated by the McTaggart A-series enhancement $X_{\pm}$ for the Allen relations. We can glue together any number of granularities by forming inverse limits, but arguably it is the finite approximations that we can process (and manipulate) — not the infinite objects (such as the real numbers) that arise at the limit.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the Association for Computing Machinery* **26** (11):832–843.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford.

E. Allen Emerson. 1990. Temporal and modal logic. In J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, pp. 995–1072. MIT Press.

Tim Fernando. 2004. A finite-state approach to events in natural language semantics. *Journal of Logic and Computation* **14** (1):79–92.

Tim Fernando. 2011. Regular relations for temporal propositions. *Natural Language Engineering* **17** (2):163–184.

Christian Freksa. 1992. Temporal reasoning based on semi-intervals. *Artificial Intelligence* **54** (11):199–227.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.

Lauri Karttunen. 2005. The Yale Shooting Problem. `http://www.stanford.edu/~laurik/fsmbook/examples/YaleShooting.html`

Uwe Lück. 2006. Continu'ous Time Goes by Russell. *Notre Dame J. Formal Logic* **47** (3):397–434.

Inderjeet Mani. 2007. Chronoscopes: a theory of underspecified temporal representations. In F. Schilder et al. (eds.) *Reasoning about Time and Events*, pp. 127–139. Springer LNAI 4795.

John M.E. McTaggart. 1908. The Unreality of Time. *Mind* **17** 457–473.

Jyrki Niemi and Kimmo Koskenniemi. 2009. Representing and Combining Calendar Information by Using Finite-State Transducers. In J. Piskorski, B. Watson and A. Yli-Jyr (eds.), *Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008*, pp. 122–133. IOS Press, Amsterdam.