Intervals and Events with and without Points

Tim Fernando*

Computer Science Department, Trinity College Dublin
Tim.Fernando@tcd.ie

Abstract. Intervals and events are examined in terms of strings with and without points, represented by symbols that occur uniquely in strings. Allen interval relations, Dowty's aspect hypothesis and inertia are understood against strings, compressed into canonical forms, describable in Monadic Second-Order logic. That understanding is built around a translation of strings that replaces stative predicates by their borders.

1 Introduction

To analyze temporal relations between events, James Allen treats intervals as primitive (not unlike [7]), noting

> There seems to be a strong intuition that, given an event, we can always "turn up the magnification" and look at its structure. ... Since the only times we consider will be times of events, it appears that we can always decompose times into subparts. Thus the formal notion of a time point, which would not be decomposable, is not useful. [1, page 834].

In place of an indivisible point, an arbitrarily decomposable interval t might be conceived as a box filled by a predicate such as *rain* that is *homogeneous* in that it holds of t iff it holds of any pair of intervals whose union is t, illustrated by the equivalence between (a) and (b).

(a) It rained from 8 am to midnight.
(b) It rained from 8 am to noon, and from 10 am to midnight.

David Dowty has famously hypothesized that

> the different aspectual properties of the various kinds of verbs can be explained by postulating a single homogeneous class of predicates — *stative predicates* — plus three or four sentential operators or connectives. [2, page 71].

Dowty's investigation of his hypothesis in terms of intervals and worlds is reformulated in [3] using strings of finite sets of homogeneous predicates. A simple

* My thanks to three anonymous referees for helpful comments.

example of such a string is the representation of the Allen *overlap* relation between intervals a and a' as the string

$$\boxed{}\boxed{a}\boxed{a,a'}\boxed{a'}\boxed{} \tag{1}$$

of length 5, starting with an empty box $\boxed{}$ for times before a, followed by $\boxed{a}$ for times in a but not a', followed by $\boxed{a,a'}$ for times in a and a', followed by $\boxed{a'}$ for times in a' but not a, followed by $\boxed{}$ for times after a'.[1] In (1), the intervals a and a' are identified with predicates interpreted as the subsets

$$U_a = \{2,3\} \quad \text{and} \quad U_{a'} = \{3,4\}$$

of the set $\{1,2,3,4,5\}$ of string positions where a and a' (respectively) occur.

In general, a string $s = \alpha_1 \cdots \alpha_n$ of n subsets α_i of a set A specifies for each $a \in A$, a subset of the set

$$[n] := \{1, \ldots, n\}$$

of string positions, namely, the set

$$U_a \ := \ \{i \in [n] \mid a \in \alpha_i\}$$

of positions where a occurs. If we repackage s as the model

$$\mathit{Mod}(s) \ := \ \langle [n], S_n, \{U_a\}_{a \in A} \rangle$$

over $[n]$ with successor relation

$$S_n \ := \ \{(i, i+1) \mid i \in [n-1]\}$$

then a theorem due to Büchi, Elgot and Trakhtenbrot says the regular languages over the set 2^A of subsets of A are given by the sentences φ of MSO_A as

$$\{s \in (2^A)^+ \mid \mathit{Mod}(s) \models \varphi\}$$

where MSO_A is Monadic Second-Order logic over strings with unary predicates labelled by A (e.g., [8]). The Büchi-Elgot-Trakhthenbrot theorem is usually formulated for strings over the alphabet A (as opposed to 2^A above), but there are at least two advantages in using the alphabet 2^A. First, for applications such as (1), it is convenient to put zero, one or more intervals in boxes for a simple temporal construal of succession. Second, for any subset $A' \subseteq A$ of A, a string $s = \alpha_1 \cdots \alpha_n \in (2^A)^+$ need only be intersected componentwise with A' to capture the *A'-reduct of $\mathit{Mod}(s)$* by the string

$$\rho_{A'}(\alpha_1 \cdots \alpha_n) \ := \ (\alpha_1 \cap A') \cdots (\alpha_n \cap A').$$

[1] Boxes are drawn instead of $\emptyset$ and curly braces $\{\cdot\}$ so as not to confuse, for example, the empty language $\emptyset$ with the string $\square$ of length one.

For example, returning to (1) with $A' = \{a\}$,

$$\rho_{\{a\}}(\boxed{\,}\boxed{a}\boxed{a,a'}\boxed{a'}\boxed{\,}) \;=\; \boxed{\,}\boxed{a}\boxed{a}\boxed{\,}\boxed{\,}.$$

Only elements of A' are observable in A'-reducts. To expand what can be observed (and turn up, as Allen puts it, the magnification), A must be enlarged (not reduced). On (1), for instance, a third interval a'' may come into view, overlapping both a and a', as depicted by the string

$$\boxed{\,}\boxed{a''}\boxed{a,a''}\boxed{a,a',a''}\boxed{a,a'}\boxed{a'}\boxed{\,}.$$

Its $\{a\}$-reduct

$$\rho_{\{a\}}(\boxed{\,}\boxed{a''}\boxed{a,a''}\boxed{a,a',a''}\boxed{a,a'}\boxed{a'}\boxed{\,}) \;=\; \boxed{\,}\boxed{\,}\boxed{a}\boxed{a}\boxed{a}\boxed{\,}\boxed{\,}$$

is, like the $\{a\}$-reduct $\boxed{\,}\boxed{a}\boxed{a}\boxed{\,}\boxed{\,}$ of (1), just another representation of $\boxed{\,}\boxed{a}\boxed{\,}$ inasmuch as any box α of homogeneous predicates is decomposable to α^n for any positive integer n. With this in mind, let us define a *stutter of* a string $\alpha_1 \cdots \alpha_n$ to be a box α_i such that $\alpha_i = \alpha_{i+1}$. To remove stutters, we apply *block compression* bc, defined by induction on the string length n

$$bc(\alpha_1 \cdots \alpha_n) \;:=\; \begin{cases} \alpha_1 \cdots \alpha_n & \text{if } n < 2 \\ bc(\alpha_2 \cdots \alpha_n) & \text{else if } \alpha_1 = \alpha_2 \\ \alpha_1 \; bc(\alpha_2 \cdots \alpha_n) & \text{otherwise} \end{cases}$$

so that $bc(s)$ has no stutter, and

$$s \text{ has no stutter} \quad \Longleftrightarrow \quad s = bc(s).$$

The finite-state approach to temporality in [4, 5] identifies a string s of sets of homogeneous predicates with its stutterless form $bc(s)$.

But can we assume a string representing an event is built solely from homogeneous predicates? It is not clear such an assumption can be taken for granted. The event nucleus of [9], for instance, postulates not only states but also events that can be extended or atomic, including points. Given a string s over the alphabet 2^A, let us agree an element $a \in A$ is an *s-point* if it occurs exactly once in s — i.e.,

$$\rho_{\{a\}}(s) \;\in\; \boxed{\,}^{*}\boxed{a}\boxed{\,}^{*} \tag{2}$$

Just as a string of statives can be compressed by removing stutters through bc, a string s of points can be compressed by deleting all occurrences in s of the empty box $\square$ for $d_\square(s)$. Line (2) above simplifies to the equation

$$d_\square(\rho_{\{a\}}(s)) \;=\; \boxed{a}.$$

We shall see that for an s-interval a, the corresponding equation is

$$d_\square(b(\rho_{\{a\}}(s))) \;=\; \boxed{l(a)}\boxed{r(a)}$$

for a certain function b on strings that associates a with a left border $l(a)$ and right border $r(a)$. The precise details are spelled out in section 2, where the set of interval relations from [1] are analyzed from the perspective of MSO_A through formulas such as

$$(\forall z)(P_a(z) \equiv (x < z \wedge z \leq y))$$

saying a occurs exactly at positions $> x$ and $\leq y$ (where P_a is the unary predicate symbol in MSO_A labeled by a, and $<$ and $\leq$ are defined from the successor relation via monadic second-order quantification). Applications to events are taken up in section 3, where the set of predicate labels is expanded in a constrained manner and the map b is inverted to expose a notion of inertia and force. A synthesis of bc and $d_{\square}$ is presented, suited to strings with or without points.

2 Strings of Points and Allen Relations

The key notion in this section is projection between strings, for which it is useful to define the *vocabulary* of a string $\alpha_1 \cdots \alpha_n$ of sets α_i to be the union

$$voc(\alpha_1 \cdots \alpha_n) \ := \ \bigcup_{i=1}^{n} \alpha_i$$

(making $voc(s)$ the $\subseteq$-least set A such that $s \in (2^A)^*$). We can then say *s projects to s'* if deleting all occurrences of the empty box $\square$ from the $voc(s')$-reduct of s yields s'

$$d_{\square}(\rho_{voc(s')}(s)) = s'$$

(recalling that $d_{\square}(\alpha_1 \cdots \alpha_n)$ is what remains after deleting each $\alpha_i = \square$). The $\mathrm{MSO}_{\{a\}}$-sentence

$$(\exists x)(\forall y)(P_a(y) \equiv x = y)$$

states there is a position where a occurs and nowhere else, as asserted in (2). It follows immediately that

Proposition 1. *The following are equivalent, for any $a \in A$ and $s \in (2^A)^*$.*

(i) $Mod(s) \models (\exists x)(\forall y)(P_a(y) \equiv x = y)$
(ii) $\rho_{\{a\}}(s) \in \square^* \boxed{a} \square^*$
(iii) *s projects to* $\boxed{a}$

Turning to (bounded) intervals, we define the string function b mentioned in the introduction relative to a set A, with which we associate a set

$$A_{\bullet} \ := \ \{l(a) \mid a \in A\} \cup \{r(a) \mid a \in A\}$$

formed from two 1-1 functions l and r, under the assumption that the three sets

$$A, \quad \{l(a) \mid a \in A\} \ \text{ and } \{r(a) \mid a \in A\}$$

are pairwise disjoint. (Think of $l(a)$ and $r(a)$ as terms — bits of syntax — rather than say, numbers.) Now, let the function

$$b_A : (2^A)^* \to (2^{A_\bullet})^*$$

map a string $\alpha_1 \cdots \alpha_n$ of subsets α_i of A to a string $\beta_1 \cdots \beta_n$ of subsets β_i of $A_\bullet$ as follows

$$\beta_i \ := \ \begin{cases} \{r(a) \mid a \in \alpha_n\} & \text{if } i = n \\ \{l(a) \mid a \in \alpha_{i+1} - \alpha_i\} \ \cup \ \{r(a) \mid a \in \alpha_i - \alpha_{i+1}\} & \text{if } i < n. \end{cases}$$

For example, for $a, a' \in A$,

$$b_A(\boxed{\ }\boxed{a}\boxed{a, a'}\boxed{a'}\boxed{\ }) \ = \ \boxed{l(a)}\boxed{l(a')}\boxed{r(a)}\boxed{r(a')}\boxed{\ } \ .$$

To simplify notation, we will often drop the subscript A on b_A. The idea behind b is that P_a is an interval iff it is the half-open interval $(l(a), r(a)]$ with open left border $l(a)$ and closed right border $r(a)$. For an interval analog of Proposition 1, recall the MSO-formula

$$\text{bounded}_a(x, y) \ := \ (\forall z)(P_a(z) \equiv (x < z \wedge z \leq y))$$

mentioned in the introduction, and observe that

$$b(\rho_{\{a\}}(s)) \ = \ \rho_{\{l(a), r(a)\}}(b(s)).$$

Proposition 2. *The following are equivalent, for any $a \in A$ and $s \in (2^A)^*$.*

(i) $Mod(s) \models (\exists x)(\exists y)(x < y \wedge bounded_a(x, y))$
(ii) $\rho_{\{a\}}(s) \in \boxed{\ }^+ \boxed{a}^+ \boxed{\ }^*$
(iii) $b(\rho_{\{a\}}(s)) \in \boxed{\ }^* \boxed{l(a)}\boxed{\ }^* \boxed{r(a)}\boxed{\ }^*$
(iv) $b(s)$ *projects to* $\boxed{l(a)}\boxed{r(a)}$

Focussing on strings s over the alphabet $2^{A_\bullet}$ (as opposed to 2^A), let us agree that a is an *s-interval* if s projects to $\boxed{l(a)}\boxed{r(a)}$ (as Proposition 2 suggests), and also that s *demarcates* A if each $a \in A$ is an s-interval. We show next how to generate the strings that demarcate a finite set A. The plan is to map any finite sequence $a_1 \cdots a_n$ into a finite set $\mathcal{L}_\bullet(a_1 \cdots a_n)$ of strings establishing

Proposition 3. *For any n distinct $a_1, \ldots, a_n$, a string s demarcates $\{a_1, \ldots, a_n\}$ iff s projects to some string in $\mathcal{L}_\bullet(a_1 \cdots a_n)$.*

The languages $\mathcal{L}_\bullet(a_1 \cdots a_n)$ are defined by induction on n. Writing ϵ for the string of length 0, we set

$$\mathcal{L}_\bullet(\epsilon) \ := \ \epsilon$$

conflating a string s as usual with the language $\{s\}$. The inductive step is

$$\mathcal{L}_\bullet(a_1 \cdots a_n a_{n+1}) \;:=\; \mathcal{L}_\bullet(a_1 \cdots a_n) \;\&^\Box\; \boxed{l(a_{n+1})}\boxed{r(a_{n+1})}$$

for a certain operation $\&^\Box$ defined as follows. Given two strings $\alpha_1 \cdots \alpha_k$ and $\alpha'_1 \cdots \alpha'_k$ of sets with the same length k, we form their componenwise union for their superposition

$$(\alpha_1 \cdots \alpha_k) \;\&\; (\alpha'_1 \cdots \alpha'_k) \;:=\; (\alpha_1 \cup \alpha'_1) \cdots (\alpha_k \cup \alpha'_k).$$

We lift & to languages L and L' stringwise

$$L \;\&\; L' \;:=\; \bigcup_{k \geq 0} \{s\&s' \mid s \in L_k \text{ and } s' \in L'_k\}$$

where L_k is the set of strings in L of length k, and similarly for L'_k. Next, we collect strings $d_\Box$-equivalent to s and s' in $d_\Box^{-1}d_\Box(s)$ and $d_\Box^{-1}d_\Box(s')$ respectively, which we superpose for

$$s \;\&_\Box\; s' \;:=\; d_\Box^{-1}d_\Box(s) \;\;\&\;\; d_\Box^{-1}d_\Box(s')$$

and then reduce to the finite set

$$s \;\&^\Box\; s' \;:=\; \{d_\Box(s'') \mid s'' \in s \;\&_\Box s'\}$$

and finally lift to languages L, L' stringwise

$$L \;\&^\Box\; L' \;:=\; \bigcup_{s \in L} \bigcup_{s' \in L'} s \;\&^\Box s'.$$

Proposition 3 is proved by induction on $n \geq 1$. The case $n = 1$ is immediate

$$\mathcal{L}_\bullet(a_1) \;=\; \epsilon \;\&^\Box\; \boxed{l(a_1)}\boxed{r(a_1)} \;=\; \boxed{l(a_1)}\boxed{r(a_1)}\,.$$

For the inductive step $n + 1$, appeal to $a_{n+1} \notin \{a_1, \ldots, a_n\}$, the induction hypothesis, and

Lemma 4. *If $voc(s) \cap voc(s') = \emptyset$, then every string in $s\&^\Box s'$ projects to $d_\Box(s)$ and $d_\Box(s)'$.*

When $a \neq a'$, a routine calculation shows

$$\mathcal{L}_\bullet(aa') \;=\; \{\mathfrak{s}_R(a, a') \mid R \in \mathcal{AR}\}$$

where the 13 interval relations in [1] constitute the set

$$\mathcal{AR} \;:=\; \{<, >, \text{d,di,f,fi,m,mi,o,oi,s,si}, =\}$$

and for each $R \in \mathcal{AR}$, $\mathfrak{s}_R(a, a')$ is the string with vocabulary

$$\{l(a), r(a), l(a'), r(a')\}$$

given in Table 1 such that for $s \in (2^A)^*$ (as opposed to $(2^{A_\bullet})^*$),

$$s \models aRa' \iff b(s) \text{ projects to } \mathfrak{s}_R(a, a').$$

R	aRa'	$\mathfrak{s}_R(a,a')$	R^{-1}	$\mathfrak{s}_{R^{-1}}(a,a')$
$<$	a before a'	$\boxed{l(a)}\boxed{r(a)}\boxed{l(a')}\boxed{r(a')}$	$>$	$\boxed{l(a')}\boxed{r(a')}\boxed{l(a)}\boxed{r(a)}$
m	a meets a'	$\boxed{l(a)}\boxed{r(a), l(a')}\boxed{r(a')}$	mi	$\boxed{l(a')}\boxed{r(a'), l(a)}\boxed{r(a)}$
o	a overlaps a'	$\boxed{l(a)}\boxed{l(a')}\boxed{r(a)}\boxed{r(a')}$	oi	$\boxed{l(a')}\boxed{l(a)}\boxed{r(a')}\boxed{r(a)}$
s	a starts a'	$\boxed{l(a), l(a')}\boxed{r(a)}\boxed{r(a')}$	si	$\boxed{l(a), l(a')}\boxed{r(a')}\boxed{r(a)}$
d	a during a'	$\boxed{l(a')}\boxed{l(a)}\boxed{r(a)}\boxed{r(a')}$	di	$\boxed{l(a)}\boxed{l(a')}\boxed{r(a')}\boxed{r(a)}$
f	a finishes a'	$\boxed{l(a')}\boxed{l(a)}\boxed{r(a), r(a')}$	fi	$\boxed{l(a)}\boxed{l(a')}\boxed{r(a), r(a')}$
$=$	a equal a'	$\boxed{l(a), l(a')}\boxed{r(a), r(a')}$	$=$	

Table 1. Allen interval relations as strings of points[2]

Given a set A of interval names and a specification $f : (A \times A) \to 2^{\mathcal{AR}}$ of sets $f(a, a')$ of Allen relations possible for pairs (a, a') from A, is there a string s that meets that specification in the sense of (3) below?

$$\text{for all } a, a' \in A, \text{ there exists } R \in f(a, a') \text{ such that } s \models aRa' \tag{3}$$

A popular tool from [1] is the transitivity table $T : (\mathcal{AR} \times \mathcal{AR}) \to 2^{\mathcal{AR}}$ mapping a pair (R, R') from $\mathcal{AR}$ to the set $T(R, R')$ of relations $R'' \in \mathcal{AR}$ such that for some intervals X, Y and Z,

$$\text{X } R \text{ Y and Y } R' \text{ Z and X } R'' \text{ Z.}$$

A function $f : (A \times A) \to 2^{\mathcal{AR}}$ is a *T-consistent labeling of A* if for all $a, a', a'' \in A$,

$$f(a, a'') \subseteq \bigcup_{R \in f(a,a')} \; \bigcup_{R' \in f(a',a'')} T(R, R').$$

T-consistency falls short of true consistency; Figure 5 in [1, page 838] provides a T-consistent labelling f of a set A of 4 intervals for which there is *no* string s of subsets of A satisfying (3) above. But for A of 3 or fewer intervals, every T-consistent labeling of A has a string s making (3) true. By Proposition 3, we can compute $T(R, R')$ by searching the language $\mathcal{L}_\bullet(a_1a_2a_3)$ for strings that satisfy a_1Ra_2 and $a_2R'a_3$

$$\begin{aligned} T(R, R') \;=\; \{R'' \in \mathcal{AR} \mid (\exists s \in \mathcal{L}_\bullet(a_1a_2a_3))\ & s \text{ projects to} \\ & \mathfrak{s}_R(a_1, a_2),\ \mathfrak{s}_{R'}(a_2, a_3) \text{ and } \mathfrak{s}_{R''}(a_1, a_3)\}\ . \end{aligned} \tag{4}$$

Implicit in (4) is a generate-and-test approach, which we can improve by refining the superposition $\&^\square$ underlying $\mathcal{L}_\bullet(a_1 \cdots a_n)$ to an operation $\&_p$ such that for all $s, s' \in (2^A - \{\square\})^*$,

$$s \;\&_p\; s' \;=\; \{s'' \in (s \;\&^\square\; s') \mid s'' \text{ projects to } s \text{ and } s'\} \tag{5}$$

[2] The strings $\mathfrak{s}_R(a, a')$ can be derived from strings $\mathfrak{s}^\circ_R(a, a')$ over the alphabet $\{a, a'\}$ by the equation $\mathfrak{s}_R(a, a') = b(\square\mathfrak{s}^\circ_R(a, a'))$. For example, $\mathfrak{s}^\circ_<(a, a') = \boxed{a}\boxed{}\boxed{a'}$ and $\mathfrak{s}^\circ_{\mathrm{m}}(a, a') = \boxed{a}\boxed{a'}$. A full list of $\mathfrak{s}^\circ_R(a, a')$, for every Allen relation R, can be found in Table 7.1 in [4, page 223].

(noting from Lemma 4 that for strings in the superposition of s with s' to project to $d_{\Box}(s)$ and $d_{\Box}(s')$, we assumed s and s' have disjoint vocabularies). To define $\&_p$, we first construct a family of 3-ary relations $\&_{\Sigma,\Sigma',\Sigma''}$ on strings over the alphabet 2^{Σ}, indexed by subsets Σ' and Σ'' of Σ. We proceed by induction, with base case

$$\frac{}{\&_{\Sigma,\Sigma',\Sigma''}(\epsilon,\epsilon,\epsilon)}$$

superposing ϵ with itself to get itself, and 3 rules which given $\&_{\Sigma,\Sigma',\Sigma''}(s,s',s'')$, extend s'' by a symbol added to s

$$\frac{\&_{\Sigma,\Sigma',\Sigma''}(s,s',s'') \qquad \alpha \subseteq \Sigma - \Sigma''}{\&_{\Sigma,\Sigma',\Sigma''}(\alpha s, s', \alpha s'')}$$

or to s'

$$\frac{\&_{\Sigma,\Sigma',\Sigma''}(s,s',s'') \qquad \alpha' \subseteq \Sigma - \Sigma'}{\&_{\Sigma,\Sigma',\Sigma''}(s, \alpha' s', \alpha' s'')}$$

or to both (in part)

$$\frac{\&_{\Sigma,\Sigma',\Sigma''}(s,s',s'') \qquad \alpha,\alpha' \subseteq \Sigma \qquad \alpha \cap \Sigma'' \subseteq \alpha' \qquad \alpha' \cap \Sigma' \subseteq \alpha}{\&_{\Sigma,\Sigma',\Sigma''}(\alpha s, \alpha' s, (\alpha \cup \alpha') s'')}$$

subject in all cases to certain conditions on the symbol added, expressed through $\Sigma, \Sigma', \Sigma''$. The case $\Sigma' = \Sigma'' = \Box$ gives $\&^{\Box}$

$$\&_{\Sigma,\Box,\Box}(s,s',s'') \iff s'' \in (s \,\&^{\Box}\, s')$$

for all $s, s', s'' \in (2^{\Sigma})^*$. More generally, however, the point of non-empty Σ' and Σ'' is to constrain the superposition according to

Proposition 5. *For all $\Sigma', \Sigma'' \subseteq \Sigma$ and $s, s', s'' \in (2^{\Sigma} - \{\Box\})^*$,*

$$\begin{aligned} \&_{\Sigma,\Sigma',\Sigma''}(s,s',s'') \iff\ & s'' \in s\&^{\Box}s' \text{ and} \\ & s'' \text{ projects to } d_{\Box}(\rho_{\Sigma'}(s)) \text{ and } d_{\Box}(\rho_{\Sigma''}(s')). \end{aligned}$$

Now, for $\&_P$, let Σ be $A_\bullet$, and Σ' be the vocabulary of s, and Σ'' be the vocabulary of s'

$$\&_P(s,s') \ := \ \{s'' \mid \&_{A_\bullet, voc(s), voc(s')}(s,s',s'')\}.$$

By Proposition 5, (5) holds for all $s, s' \in (2^{A_\bullet} - \{\Box\})^*$. We can then sharpen the computation of $T(R,R')$ by (4) to the set of relations $R'' \in \mathcal{AR}$ such that

$$\mathfrak{s}_R(a_1,a_2) \ \&_P \ \mathfrak{s}_{R'}(a_2,a_3) \text{ has a string that projects to } \mathfrak{s}_{R''}(a_1,a_3).$$

Also, to check if a labeling f of A that specifies singleton sets $\{R^{a,a'}\} = f(a,a')$ has a string satisfying (3), we $\&_P$-superpose together each $\mathfrak{s}_{R^{a,a'}}(a,a')$. Apart from transitivity tables and (3), $\&_P$ applies to the constrained generation of strings in or out of $\mathcal{L}_\bullet(a_1 \cdots a_n)$, with projection constraints beyond intervals.

3 Expansions, Inertia and Events

The requirement that $l(a)$ and $r(a)$ mark the left and right borders of a can be expressed with the help of certain $\mathrm{MSO}_{\{a\}}$-formulas over a free variable x. Let $\chi_{l(a)}(x)$ say P_a holds right after x but not at x

$$\chi_{l(a)}(x) \ := \ \neg P_a(x) \wedge (\exists y)(xSy \wedge P_a(y))$$

and $\chi_{r(a)}(x)$ say P_a holds at x but not right after

$$\chi_{r(a)}(x) \ := \ P_a(x) \wedge \neg(\exists y)(xSy \wedge P_a(y)).$$

We can then interpret $P_{l(a)}$ and $P_{r(a)}$ in terms of P_a according to the set

$$\Phi(A_\bullet) \ := \ \{(\forall x)(P_t(x) \equiv \chi_t(x)) \mid t \in A_\bullet\}$$

of $\mathrm{MSO}_{A \cup A_\bullet}$-sentences equating $P_t(x)$ with $\chi_t(x)$. Given a string s of subsets of $A_\bullet$, the strings over the alphabet 2^A that b maps to s can be collected in the set

$$b^{-1}s \ = \ \{\rho_A(s') \mid s' \in s\&(2^A)^* \text{ and } (\forall \varphi \in \Phi(A_\bullet)) \ \mathit{Mod}(s') \models \varphi\}$$

in 3 steps

Step 1: expand with labels from A, superposing s with $(2^A)^*$
Step 2: constrain by $\Phi(A_\bullet)$
Step 3: reduce by ρ_A (for A-reduct).

To compute b from Steps 1-3 above, it suffices to replace A by $A_\bullet$ in steps 1 and 3. The difference between b and b^{-1} comes down to the subalphabet added in Step 1 and preserved in Step 3. $\Phi(A_\bullet)$ is, however, arguably more in sync with b than with b^{-1}, grounding, as it does, $P_{l(a)}$ and $P_{r(a)}$ in P_a. The inverse b^{-1} invites us to consider the reverse:

(Q) how do we interpret P_a, given interpretations of $P_{l(a)}$ and $P_{r(a)}$?

Answering (Q) is an instructive exercise, pointing to forces and events.

Our answer to (Q) comes in two parts, assuming $P_{l(a)}$ and $P_{r(a)}$ are interpreted as subsets $U_{l(a)}$ and $U_{r(a)}$ (respectively) of the set $[n]$ of positions of a string of length n. The first part is an inductive construction

$$U_a = \bigcup_{i \geq 0} U_{a,i} \tag{6}$$

of the interpretation U_a of P_a according to the definitions

$$\begin{aligned} U_{a,0} \ &:= \ U_{r(a)} \\ U_{a,i+1} \ &:= \ U_{a,i} \cup \{k \in [n-1] \mid k+1 \in U_{a,i} \text{ and } k \notin U_{l(a)}\} \end{aligned}$$

suggested by the implications

$$\begin{aligned}(\forall x)(P_{r(a)}(x) \supset P_a(x))\\ (\forall x)(\forall y)((xSy \wedge P_a(y) \wedge \neg P_{l(a)}(x)) \supset P_a(x))\end{aligned} \tag{7}$$

from $\Phi(A_\bullet)$. The second part of our answer consists of two conditions

$$U_{l(a)} \cap U_a = \emptyset \tag{8}$$

$$\{i+1 \mid i \in U_{l(a)}\} \subseteq U_a \tag{9}$$

expressed by the implications

$$\begin{aligned}&(\forall x)(P_{l(a)}(x) \supset \neg P_a(x))\\ &(\forall x)(P_{l(a)}(x) \supset (\exists y)(xSy \wedge P_a(y)))\end{aligned}$$

implicit in $\Phi(A_\bullet)$. (8) and (9) hold precisely if $l(a)$ and $r(a)$ are positioned properly under $U_{l(a)}$ and $U_{r(a)}$ — i.e., there is a string in

$$(\epsilon + \boxed{r(a)})(\boxed{l(a)}\boxed{r(a)})^*$$

to which the string s corresponding to the MSO$_{A_\bullet}$-model $\langle [n], S_n, \{U_{t\in A_\bullet}\}\rangle$ projects

$$d_\Box(\rho_{\{l(a),r(a)\}}(s)) \in (\epsilon + \boxed{r(a)})(\boxed{l(a)}\boxed{r(a)})^*. \tag{10}$$

Proposition 6. *For every $s \in (2^{A_\bullet})^*$,*

$$b^{-1}s \neq \emptyset \iff \text{(10) holds for every } a \in A.$$

Moreover, if $Mod(s) = \langle [n], S_n, \{U_t\}_{t\in A_\bullet}\rangle$ then for every $s' \in (2^A)^*$ such that $b(s') = s$,

$$Mod(s') = \langle [n], S_n, \{U_a\}_{a\in A}\rangle$$

where U_a is given by (6) above from the sets $U_{l(a)}$ and $U_{r(a)}$ in $Mod(s)$. That is, under b, P_a is definable from $P_{l(a)}$ and $P_{r(a)}$ according to

$$P_a(x) \ \equiv (\exists X)(X(x) \wedge a\text{-}path(X))$$

where a-$path(X)$ is the conjunction

$$\forall x(X(x) \supset P_{r(a)}(x) \vee \exists y(xSy \wedge X(y))\ \wedge\ \neg\exists x(X(x) \wedge P_{l(a)}(x))$$

saying X is an S-path to $P_{r(a)}$ that avoids $P_{l(a)}$.

A crucial ingredient of the analysis of b^{-1} described by Proposition 6 is the implication (7) underlying the inductive step $U_{a,i+1}$. That step lets a spread to the neighboring left box unless $l(a)$ is in that box. This property of $l(a)$ can be isolated in a label $f(a)$ constrained by the implication

$$(\forall x)(\neg P_a(x) \wedge (\exists y)(xSy \wedge P_a(y)) \;\supset\; P_{f(a)}(x)) \tag{11}$$

which (without the converse of (11)) falls short of reducing $P_{f(a)}$ to $P_{l(a)}$. For the sake of symmetry, we also introduce labels $\overline{a}$ and $P_{f(\overline{a})}$ subject to

$$(\forall x)(P_{\overline{a}}(x) \equiv \neg P_a(x))$$

making $\overline{a}$ the negation of a, and the implication

$$(\forall x)(P_a(x) \wedge (\exists y)(xSy \wedge \neg P_a(y)) \;\supset\; P_{f(\overline{a})}(x)). \tag{12}$$

tracking, with (11), any changes in $a/\overline{a}$. The intuition is that $f(a)$ and $f(\overline{a})$ mark the applications of forces for and against a (respectively). The slogan behind (11) and (12) is

no change unless forced

or, in one word, *inertia*. To save that principle from vacuity, let us be careful *not* to identify $f(a)$ with $l(a)$ or $f(\overline{a})$ with $r(a)$. Indeed, insofar as clashing forces are commonplace and merit logical scrutiny (rather than dismissal), there is nothing illegitimate about a box containing both $f(a)$ and $f(\overline{a})$. By contrast, $l(a)$ and $r(a)$ are mutually exclusive under (10). Over any given stretch of time, any number of forces can be at play, some of which may be neutralized by competition. A force in isolation may have very different effects with company. That said, there is no denying that we detect and evaluate forces by the state changes they effect.

Stative predicates labelled by $a \in A$ differ significantly from non-stative predicates labelled by $l(a), r(a), f(a)$ and $f(\overline{a})$ in how strings built from them compress to canonical forms. Recall from the Introduction the link between homogeneity and block compression bc, deleting stutters

$$\mathit{bc}^{-1}\mathit{bc}(s) = \alpha_1^+ \cdots \alpha_n^+ \quad \text{if } \mathit{bc}(s) = \alpha_1 \cdots \alpha_n$$

just as $d_\square$ deletes $\square$

$$d_\square^{-1} d_\square(s) = \square^* \alpha_1 \square^* \cdots \square^* \alpha_n \square^* \quad \text{if } d_\square(s) = \alpha_1 \cdots \alpha_n.$$

Proposition 7. *For every $s \in (2^A)^*\square$, $b(\mathit{bc}(s)) = d_\square(b(s))\square$ and $\mathit{bc}(s)$ is the unique string over 2^A in the set $b^{-1}(d_\square(b(s))\square)$.*[3]

Underlying both notions of compression, $d_\square$ and bc, is the slogan

[3] $\square$ is put after $(2^A)^*$ and after $d_\square(b(s))$ to reconcile a difference between a's left and right borders, $l(a)$ and $r(a)$; whereas $r(a)$ is in a, $l(a)$ is outside a. This gives rise to a wrinkle in Proposition 2, line (ii), $\rho_{\{a\}}(s) \in \square^+\boxed{a}^+\square^*$. The Kleene star $\square^*$ becomes a plus in Proposition 7, with $s \in (2^A)^*\square$ necessitating a $\square$ after $d_\square(b(s))$.

no time without change.

But while bc represents that change in terms of decomposable intervals/statives, $d_\Box$ employs non-decomposable points/borders (not to mention forces). The function $d_\Box$ is simpler than bc, and provides a pointed twist on the interval-based approach in [3] to Dowty's aspect hypothesis.

An obvious question is how to compress a string s of sets consisting of labels for stative and non-stative predicates alike (as in Step 2 above). Let us collect labels of non-homogeneous predicates in a set C. A simple synthesis of $d_\Box$ and bc is the function $bc^C(s)$ defined by induction on the length of s as follows. Let $bc^C(\epsilon) := \epsilon$ and

$$bc^C(\alpha s) \ := \ \begin{cases} bc^C(s) & \text{if } \alpha = \Box \text{ or } (\alpha \cap C = \emptyset \text{ and } s \text{ begins with } \alpha) \\ \alpha\, bc^C(s) & \text{otherwise.} \end{cases}$$

It is easy to see that

$$bc^C(s) \ = \ d_\Box(s) \ \ \text{if } s \in (2^C)^*$$

while for any label ☺ $\not\in C \cup voc(s)$,

$$bc(s) \ = \ d_☺(bc^C(i_☺(s))) \ \ \text{if } voc(s) \cap C = \emptyset$$

where $i_☺$ inserts ☺

$$i_☺(\alpha_1 \cdots \alpha_n) \ := \ (\alpha_1 \cup \{☺\}) \cdots (\alpha_n \cup \{☺\})$$

and $d_☺$ deletes ☺

$$d_☺(\alpha_1 \cdots \alpha_n) \ := \ (\alpha_1 - \{☺\}) \cdots (\alpha_n - \{☺\}).$$

Any void $\Box$ is filled with ambient noise ☺, which we may otherwise ignore.

4 Conclusion

We can summarize the work above as follows, recalling the passages from [1] and [2] quoted in the Introduction. We "turn up the magnification" by inverting A-reducts ρ_A, and analyze homogeneous statives through block compression bc, reconstructed according to Proposition 7 through a border translation b and $\Box$-removal $d_\Box$. Working with strings s, we form A-canonical representations by compressing $\rho_A(s)$ according to

$$s\Box s' \ \approx \ ss' \quad (\text{leading to } d_\Box)$$

and

$$s\alpha\alpha s' \ \approx \ s\alpha s' \ \text{ if } \alpha \cap C = \emptyset \quad (\text{leading to } bc)$$

where C collects labels of non-homogeneous predicates (including forces). Among the labels in C are s-points that (as defined in section 2) describe *particulars*, in contrast to labels that occur in more than one position in s (describing *universals*). Handling granularity through A and A-reducts is a hallmark of *institutions* ([6]), where models and sentences are organized around *signatures* for variable granularity. To view MSO as an institution, we pair a set A up with a subset B of A for a signature (A, B); a model of signature (A, B) is then a string $s \in (2^A)^*$ such that each element of B is an s-point, which serves as a first-order variable to express MSO predication in a sentence of signature (A, B). That said, the set C of non-homogeneous predicates is not restricted to s-points. Indeed, line (10) in section 3 allows $r(a)$ and $l(a)$ to occur more than once in a string s for which $b^{-1}s$ is non-empty. The advance over [4, 5] that the present work may claim has less to do with the particular notion, s-point, than with the deletion $d_\square$ of $\square$, simplifying block compression bc. Compression $d_\square$ and bc carve out two sides of a coin, the border translation b that yields s-points and more.

References

1. J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. D.R. Dowty. *Word Meaning and Montague Grammar*. Reidel, Dordrecht, 1979.
3. T. Fernando. Dowty's aspect hypothesis segmented. In *Proceedings of the 19th Amsterdam Colloquium*, pages 107–114. 2013.
4. T. Fernando. The semantics of tense and aspect: A finite-state perspective. In S. Lappin and C. Fox, editors, *Handbook of Contemporary Semantic Theory*, pages 203–236. Wiley-Blackwell, second edition, 2015.
5. T. Fernando. On regular languages over power sets. *Journal of Language Modelling*, 4(1):29–56, 2016.
6. J.A. Goguen and R.M. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.
7. C.L. Hamblin. Instants and intervals. *Studium Generale*, 24:127–134, 1971.
8. L. Libkin. *Elements of Finite Model Theory*. Springer, 2010.
9. M. Moens and M. Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15– 28, 1988.