Intensions, Types and Finite-State Truthmaking

Tim Fernando

Abstract Intensions are formulated as non-deterministic relations computed by finite-state transducers, and types as regular languages in an account at bounded but refinable granularity of the temporal structure of events. Strings representing timelines are linked to deterministic finite automata encoding argument structure and truthmakers, based on many notions of part.

1 Introduction

Formal investigations of natural language semantics descending from Carnap (1947) and Montague (1974) analyze an expression e as its (Carnap-Montague) *intension* CMI_e, a function mapping an *index* i for evaluating e to the *denotation* (extension or value) $\mathsf{CMI}_e(i)$ *of* e *at* i. For a declarative statement e, the Fregean tradition is that at a suitable index i, the denotation of e is one of two truth values, differentiating truth from falsity. Truth values are replaced by types for more refined denotations in type-theoretic approaches such as Ranta (1994), Cooper (2005) and Luo (2012) that equate the truth of e at i with inhabitation of the type $\mathsf{CMI}_e(i)$

$$e \text{ is true at } i \iff \text{the type } \mathsf{CMI}_e(i) \text{ is non-empty.} \qquad (\dagger)$$

The biconditional (†) suggests that an object of type $\mathsf{CMI}_e(i)$ is, to borrow a term from Mulligan et al. (1984) (MSS84), a *truthmaker of* e *at* i. Under the celebrated *propositions-as-types* principle identifying logical forms with types, truthmakers of conjunctions are pairs, truthmakers of implications are functions, and, in general, a sentence's logical form shapes the form of its truthmakers. This constraint on the form of truthmakers is an instance of what MSS84 calls "the dogma of logical form." MSS84 rejects the dogma, advocating instead

T. Fernando (✉)
Trinity College, Dublin, Ireland
e-mail: Tim.Fernando@tcd.ie

© Springer International Publishing AG 2017
S. Chatzikyriakidis and Z. Luo (eds.), *Modern Perspectives in Type-Theoretical Semantics*, Studies in Linguistics and Philosophy 98, DOI 10.1007/978-3-319-50422-3_9

> the independence of ontological from logical complexity: ontologically complex objects (those having proper parts) are not for that reason also in some way logically complex, any more than there is reason to suppose that to every logically complex (true) sentence there corresponds an ontologically complex entity which makes it true. [page 298]

Smith (1999) argues further that

> there is no superficial feature (for example the logical form of the corresponding sentence) which will allow us to determine in some quasi-automatic fashion the totality of all of that to which reference is made in a given judgment. [page 286]

Calling a sentence's logical form a "superficial feature" is disparaging; if work on natural language semantics has shown anything, it is the non-triviality of constructing logical forms for everyday language. Apart from what is explicit in the expression e, there is typically information implicit in an utterance u crucial to understanding e, as part of u. If that implicit information is to find its way into the type $\mathsf{CMI}_e(i)$, it is through the representation of u by the index i (serving as the context for associating e with the logical form $\mathsf{CMI}_e(i)$). Insofar as i leaves out information about u relevant to e (falling short of "the totality of all of that to which reference is made"), $\mathsf{CMI}_e(i)$ cannot be anything more than a "superficial feature" of e relative to u. The importance of representing u is arguably the key insight of *Discourse Representation Theory* (DRT, Kamp and Reyle 1993), which formulates i and $\mathsf{CMI}_e(i)$ alike as *discourse representation structures*, DRSs. From the perspective of Ranta (1994), Cooper (2005) and Luo (2012), DRSs are just types. The common practice in DRT, however, is to interpret DRSs model-theoretically (following the Montagovian custom), rather than proof-theoretically (as in constructive type theories). Indeed, a model for interpreting the DRS formulating the denotation $\mathsf{CMI}_e(i)$ of e at i might be incorporated into the index i, alongside, if not replacing, the DRS formulating i.

In the present paper, models are used as truthmakers inhabiting types, while indices specify varying levels of bounded granularity, relative to which the models are understood as approximations and can be assumed to be finite. The focus is widened from a single index to many, approached bottom-up from their various parts that serve as truthmakers. Refining the granularity may sharpen the approximations, but bounds on any fixed level of granularity keep an approximation finite. More concretely, we proceed in Sect. 2 from the example of truthmakers as events (Davidson 1967, page 91), analyzing Davidson's sentence (1) as the attribute value structure (2), which we reduce to the set (3) of six strings, against which to interpret the modal formula (4).

(1) Jones did it slowly, deliberately, in the bathroom with a knife, at midnight

(2) $$\begin{bmatrix} \mathit{agent} & = & \text{jones} \\ \mathit{how} & = & \begin{bmatrix}\text{slow} \\ \text{deliberate}\end{bmatrix} \\ \mathit{where} & = & \text{bathroom} \\ \mathit{with} & = & \text{knife} \\ \mathit{when} & = & \text{midnight} \end{bmatrix}$$

(3) $$\left\{ \begin{array}{c} \mathit{agent}\ \text{jones}, \\ \mathit{how}\ \text{slow}, \\ \mathit{how}\ \text{deliberate}, \\ \mathit{where}\ \text{bathroom}, \\ \mathit{with}\ \text{knife}, \\ \mathit{when}\ \text{midnight} \end{array} \right\}$$

(4) $\langle \mathit{agent} \rangle$ jones $\wedge$ $\langle \mathit{how} \rangle$ slow $\wedge$ $\langle \mathit{how} \rangle$ deliberate $\wedge$ $\langle \mathit{where} \rangle$ bathroom $\wedge$ $\langle \mathit{with} \rangle$ knife $\wedge$ $\langle \mathit{when} \rangle$ midnight

Temporal relations between events (including speech events useful in accounts of tense) are specified by locating them in a string s that encodes a timeline. Toward this end, we associate an event q with a set $\mathscr{L}(q)$ of strings characterizing q in that for all positions i and j in s with $i \leq j$,

$$q \text{ occurs at } (i, j) \text{ within } s \iff s_i^j \in \mathscr{L}(q)$$

where s_i^j is the substring of s from position i to j (given, for $s = \alpha_1 \cdots \alpha_n$, by $\alpha_i \alpha_{i+1} \cdots \alpha_j$). For example, we might represent q by symbols b_q and e_q specifying where q begins and ends if we let $\mathscr{L}(q)$ be the set of strings where b_q and e_q occur exactly once, namely, at the first and last string positions. More interesting examples of $\mathscr{L}(q)$ are described in Sect. 3, locating an arbitrary finite set of events within a single string s through a satisfaction relation $\models$ with formulas φ. For many formulas φ, there exist a binary "part" relation $\leq_\varphi$ between strings, and a set $T(\varphi)$ of strings that serve as truthmakers according to the biconditional (‡)

$$s \models \varphi \iff (\exists s' \leq_\varphi s)\, s' \in T(\varphi). \qquad (‡)$$

(‡) says: s satisfies φ precisely when s has a part that is a truthmaker of φ. For instance, (‡) holds with

- φ as the formula occurs(q)
- $T(\varphi)$ as the aforementioned language $\mathscr{L}(q)$, and
- $\leq_\varphi$ as the factor relation, summing over substrings s_i^j of s

$$s' \text{ factor } s \iff s' = s_i^j \text{ for some positions } i, j \text{ in } s.$$

Implicit in (‡) is a set C_φ of strings, to which s is assumed to belong, corresponding to the domain of the intension CMI_φ of φ, understood as a function mapping $s \in C_\varphi$ to the set of parts of s that are truthmakers of φ

$$\mathsf{CMI}_\varphi(s) = \{s' \in T(\varphi) \mid s' \leq_\varphi s\}$$

so that

$$s \models \varphi \iff \mathsf{CMI}_\varphi(s) \text{ is non-empty.}$$

The intuition is C_φ represents the presuppositions of φ, while $T(\varphi)$ represents the assertion at issue in φ, linked to C_φ by a part relation $\leq_\varphi$. A variety of part relations is handy in, for example, differentiating sentences made up of the same words, such as

$$\text{John loves Mary} \neq \text{Mary loves John}$$

analyzed in Sect. 2 through various notions of part for the inequivalence

$$\text{love} \wedge \langle agent\rangle\text{john} \wedge \langle patient\rangle\text{mary} \;\;\not\equiv\;\; \text{love} \wedge \langle agent\rangle\text{mary} \wedge \langle patient\rangle\text{john}.$$

Beyond multiple notions of part, it will prove useful in Sects. 2 and 3 to work with many sets C_φ of constraints, as well as sets $T(\varphi)$ of truthmakers. While varying these notions, care will be taken to keep them finite-state so that the relation

$$\{(s, s') \in C_\varphi \times T(\varphi) \mid s' \leq_\varphi s\}$$

encoding CMI_φ is computed by some finite-state transducer (in the interest of computational tractability and cognitive plausibility).

2 Attribute Value Structures as Events and States

Recalling lines (1)–(4) from the Introduction, the attribute value structure (2) for Davidson's example (1) of event modification/predication follows an established tradition in computational linguistics that has received renewed attention with the interest in frames (e.g., Fillmore 1982; Petersen 2007; Cooper 2016) as well as linking semantics (Beaver and Condoravdi 2007; Champollion 2015). The connection with modal logic illustrated by (4) is studied at length in, for example, Blackburn (1993). The interpretation against a language (i.e., set of strings) such as (3) is emphasized in Fernando (2016), from which the present section borrows freely.

2.1 Derivatives

A key idea is that given a language L and a string s, the *s-derivative of L* is the set

$$L_s \;:=\; \{s' \mid ss' \in L\}$$

of strings that put after s belong to L (Brzozowski 1964). The chain of equivalences

$$\begin{aligned} a_1a_2\cdots a_n \in L \quad &\Longleftrightarrow \quad a_2\cdots a_n \in L_{a_1} \\ &\Longleftrightarrow \quad a_3\cdots a_n \in L_{a_1a_2} \\ &\Longleftrightarrow \quad \cdots \\ &\Longleftrightarrow \quad \varepsilon \in L_{a_1\cdots a_n} \end{aligned}$$

from a string $a_1\cdots a_n$ to the empty/null string ε means that L is accepted by the deterministic automaton with

- s-derivatives L_s as states
- initial state $L = L_\varepsilon$
- a-transitions from L_s to L_{sa} (for every symbol a), and
- final states L_s such that $\varepsilon \in L_s$.

The Myhill–Nerode theorem says that a language L over a finite alphabet A is regular iff the set $\{L_s \mid s \in A^*\}$ of derivatives of L is finite (e.g., Hopcroft and Ullman 1979); indeed, the Myhill-Nerode equivalence $\sim_L$ between strings with the same continuations in L is just equality of derivatives

$$s \sim_L s' \iff L_s = L_{s'}.$$

Note that L_s is non-empty precisely if s is the prefix of some string in L. Moreover, if L_s is empty then so is L_{sa} for every symbol a. That is, $\emptyset$ is a sink state that we may safely exclude from the states of the automaton above, at the cost of making the transition function partial. Let us define an A-*state* to be a non-empty subset q of A^* that is prefix-closed (i.e., for all $sa \in q$, $s \in q$). An A-state q can then make an a-transition to its a-derivative q_a precisely if $a \in q$.

2.2 *Satisfaction*

Now, for any set X, let $Fin(X)$ be the set of finite subsets of X. Fix an infinite set Lab of labels, and for $A \in Fin(Lab)$, let $sen(A)$ be the set of formulas generated from $a \in A$ according to

$$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \langle a\rangle\varphi$$

(Hennessy and Milner 1985). We interpret these formulas over A-states q, treating $\top$ as a tautology, $\neg$ as negation, $\wedge$ as conjunction, and $\langle a\rangle$ as a diamond modal operator for a-transitions

$$q \models \langle a\rangle\varphi \iff a \in q \text{ and } q_a \models \varphi.$$

We extend $\langle a\rangle\varphi$ from $a \in A$ to strings $s \in A^*$, setting $\langle\varepsilon\rangle\varphi$ to φ, and inductively,

$$\langle as\rangle\varphi := \langle a\rangle\langle s\rangle\varphi$$

from which it follows that

$$\langle a_1 \cdots a_n\rangle\varphi = \langle a_1\rangle \cdots \langle a_n\rangle\varphi.$$

and for every A-state q,

$$q \models \langle s \rangle \varphi \iff s \in q \text{ and } q_s \models \varphi.$$

Next, given a string $s \in \mathit{Lab}^*$ and a set $A \in \mathit{Fin}(\mathit{Lab})$, we compute the longest prefix of s that belongs to A^* by the function $\pi_A : \mathit{Lab}^* \to A^*$ defined by

$$\begin{aligned} \pi_A(\varepsilon) &:= \varepsilon \\ \pi_A(as) &:= \begin{cases} a\pi_A(s) & \text{if } a \in A \\ \varepsilon & \text{otherwise.} \end{cases} \end{aligned}$$

The A-*restriction of* a language $q \subseteq \mathit{Lab}^*$ is the image of q under π_A

$$q \restriction A := \{\pi_A(s) \mid s \in q\}\ .$$

If q is an Lab-state, then its A-restriction, $q \restriction A$, is an A-state and is just the intersection $q \cap A^*$ with A^*. A-restrictions are interesting because satisfaction $\models$ of formulas in $\mathit{sen}(A)$ can be reduced to them.

Proposition 1 *For every $A \in \mathit{Fin}(\mathit{Lab})$, $\varphi \in \mathit{sen}(A)$ and Lab-state q,*

$$q \models \varphi \iff q \restriction A \models \varphi$$

and if, moreover, $s \in q \restriction A$, then

$$q \models \langle s \rangle \varphi \iff (q \restriction A)_s \models \varphi.$$

Proposition 1 is proved by a routine induction on $\varphi \in \mathit{sen}(A)$ and $s \in q \restriction A$. There is structure lurking in Proposition 1, taken up in Fernando (2016). To see the relevance to lines (1)–(4) above, let Lab contain the finite set

$$\begin{aligned} A := \{&\mathit{agent}, \mathit{how}, \mathit{where}, \mathit{with}, \mathit{when}, \\ &\text{jones, slow, deliberate, bathroom, knife, midnight}\} \end{aligned}$$

so that if q is the A-state $\{\mathit{agent}, \mathit{how}, \mathit{where}, \mathit{with}, \mathit{when}, \varepsilon\}$, then the union of q with (3) can be construed as the attribute value structure (2) of type q inasmuch as for every $s \in q$, its s-derivative is non-empty. For an account of record types (which have proved useful in linguistic semantics; e.g., Cooper and Ginzburg 2015), it is helpful to close the set $\mathit{sen}(A)$ of sentences φ under the construct $\Box_B \varphi$, for every $B \subseteq A$, with

$$q \models \Box_B \varphi \iff (\forall s \in q \cap B^*)\ q_s \models \varphi$$

for every A-state q. Proposition 1 holds with this modification to $\mathit{sen}(A)$ and $\models$.

To accommodate open-ended descriptions suggested by Davidson's sentence (1), we have left open exactly what the set *Lab* of labels is, allowing it to be infinite so that it may have arbitrarily large finite subsets. In particular, we may include in *Lab* any number of names such as jones that picks out a state

$$q \models \langle \text{jones} \rangle \top \iff q \text{ is named 'jones'}$$

just as world variables in Prior (1967) and *nominals* in Hybrid Logic (e.g. Blackburn 1993).

3 Timelines as Strings

For temporal relations between events belonging to some set Q, it will be convenient to fix a linear order $(T, <)$ such as the real line and assume each event $q \in Q$ is assigned an interval $\tau(q) \subseteq T$. (We will weaken this assumption later.) Let us map each $t \in T$ to the function $f_t : Q \to \{0, 1, 2\}$ comparing t to each interval $\tau(q)$ as follows

$$f_t(q) := \begin{cases} 0 & \text{if } (\forall t' \in \tau(q))\ t < t' \\ 1 & \text{if } t \in \tau(q) \\ 2 & \text{otherwise (i.e., } (\forall t' \in \tau(q))\ t' < t). \end{cases}$$

We partially order functions $f, f' : Q \to \{0, 1, 2\}$ componentwise

$$f \leq f' \iff (\forall q \in Q)\ f(q) \leq f'(q).$$

The map $t \mapsto f_t$ preserves $\leq$

$$t \leq t' \text{ implies } f_t \leq f_{t'}$$

whilst respecting the intervals $\tau(q)$ in that whenever $f_t = f_{t'}$ and $q \in Q$,

$$t \in \tau(q) \iff t' \in \tau(q).$$

Now, assuming Q is finite, then so is the set

$$T_\tau := \{f_t \mid t \in T\}$$

and there is a unique string $a_1 a_2 \cdots a_n \in T_\tau^+$ such that

$$T_\tau = \{a_i \mid 1 \leq i \leq n\} \quad \text{and} \quad a_1 < a_2 < \cdots < a_n$$

(where $a < a'$ abbreviates the conjunction $a \leq a'$ and $a \neq a'$). Moreover, within $a_1 \cdots a_n$, we can locate each $q \in Q$ in the image of $\tau(q)$ under $t \mapsto f_t$

$$\tau_f[q] \ := \ \{f_t \mid t \in \tau(q)\}$$

and agree that

$$q \text{ occurs at } (i, j) \text{ within } a_1 \cdots a_n \quad \Longleftrightarrow \quad \{a_k \mid i \leq k \leq j\} \ = \ \tau_f[q].$$

The existence of the string $a_1 \cdots a_n$ depends on the finiteness of Q. The length n of the string grows as we expand Q to a larger finite set, increasing the cardinality of a function $f_t : Q \to \{0, 1, 2\}$, construed as a subset of the finite set $Q \times \{0, 1, 2\}$. For any $q \in Q$ and $f : Q \to \{0, 1, 2\}$, if f^q is the subset $\{(q, f(q))\}$ of f, then clearly,

$$a_1^q \cdots a_n^q \ \in \ \{(q, 0)\}^*\{(q, 1)\}^+\{(q, 2)\}^*.$$

3.1 Fluents and Monadic Second-Order Logic

Moving away from the particular functions f_t in T_τ, let us encode a finite timeline as a string, using temporal propositions, called *fluents* for short (generalizing over the pairs (q, i) in f_t). We work with finite sets Σ of fluents for bounded granularity, enlarging Σ to lengthen the strings (refining the grain). The idea is familiar from the representations of a calendar year at various granularities. The set Σ = {Jan, Feb, . . ., Dec} of months suggests the string

$$s_\Sigma \ := \ \boxed{\text{Jan}}\boxed{\text{Feb}} \cdots \boxed{\text{Dec}}$$

of length 12; enlarging Σ with days d1,d2,. . .,d31 refines s_Σ to the string

$$\boxed{\text{Jan,d1}}\boxed{\text{Jan,d2}} \cdots \boxed{\text{Jan,d31}}\boxed{\text{Feb,d1}} \cdots \boxed{\text{Dec,d31}}$$

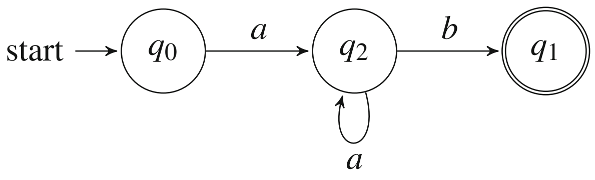
of length 366 for a leap year (drawing boxes instead of the usual curly braces { and } around sets *qua* symbols to suggest a film strip). While it is irresistible to call the boxes snapshots, a change in Σ can cause a box to split, as $\boxed{\text{Jan}}$ in s_Σ does (30 times) on adding days

$$\boxed{\text{Jan}} \ \rightsquigarrow \ \boxed{\text{Jan,d1}}\boxed{\text{Jan,d2}} \cdots \boxed{\text{Jan,d31}}.$$

Similarly, a common Reichenbachian account of the progressive puts a *reference time* R inside the event time E, splitting $\boxed{\text{E}}$ into 3 boxes

$$\boxed{\text{E}} \ \rightsquigarrow \ \boxed{\text{E}}\boxed{\text{E,R}}\boxed{\text{E}}$$

(one before, one simultaneous, and one after R). We can also encode runs of a finite automaton as strings. Take, for example, the finite automaton $\mathscr{A}$ with three transitions



over the initial state q_0 and final (or accepting) state q_1. A run of $\mathscr{A}$ is a sequence of transitions that $\mathscr{A}$ makes, such as $q_0 \xrightarrow{a} q_2 \xrightarrow{b} q_1$, in the course of which, the finite automaton $\mathscr{A}$ changes its current state from q_0 to q_2 to q_1. We can encode this run as $\boxed{a, q_2}\boxed{b, q_1}$, leaving out the automaton's initial state q_0.

Encoding runs as strings is useful for expressing the languages accepted by finite automata in Monadic Second-Order logic (MSO), one half of Büchi's theorem (e.g., (Libkin 2004), Theorem 7.21, pp 124–126). Strings are construed as models of predicate logic, associating a finite set Σ with a signature Σ_S specifying a unary relation symbol P_a, for each $a \in \Sigma$, alongside a binary relation symbol S. The intent is that S express the successor relation between string positions, and P_a pick out the positions where a occurs. For instance,

$$\exists x \exists y (S(x, y) \wedge P_a(x))$$

says a occurs before the end of the string. Confining our attention to finite models of size $n \geq 0$, we write $[n]$ for the set of integers from 1 to n

$$[n] \ := \ \{1, 2, \ldots, n\}$$

(with $[0] = \emptyset$) and S_n for the successor relation on $[n]$

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

(with $S_1 = S_0 = \emptyset$). A *Σ_S-model* is a tuple

$$M = \langle [n], S_n \{P_a^M\}_{a \in \Sigma} \rangle$$

(for some $n \geq 0$) mapping each $a \in \Sigma$ to a subset P_a^M of $[n]$. For each $i \in [n]$, let us collect all $a \in \Sigma$ such that $i \in P_a^M$ in

$$\alpha_i \ := \ \{a \in \Sigma \mid i \in P_a^M\}.$$

There is a bijection between Σ_S-models M and strings $\alpha_1 \cdots \alpha_n$ over the powerset 2^{Σ} of Σ, as the equivalence

$$i \in P_a^M \iff a \in \alpha_i$$

goes from M to $\alpha_1 \cdots \alpha_n$ and back. That is, a string $\alpha_1 \cdots \alpha_n \in (2^{\Sigma})^*$ can serve as a Σ_S-model against which to interpret predicate logic formulas such as $\exists x \exists y (S(x, y) \wedge P_a(x))$. We can add any finite number of fluents to Σ, working with the set $Fin(\Theta)$ of finite subsets of some fixed large set Θ of fluents, including second-order and first-order variables (the latter of which are constrained to occur at exactly one position in strings representing MSO-models and valuations). A string $\alpha_1 \cdots \alpha_n$ of subsets α_i of Θ has a natural projection in $(2^{\Sigma})^n$ given by componentwise intersection with Σ

$$\rho_{\Sigma}(\alpha_1 \cdots \alpha_n) \ := \ (\alpha_1 \cap \Sigma) \cdots (\alpha_n \cap \Sigma)$$

defining a function $\rho_{\Sigma} : (2^{\Theta})^* \to (2^{\Sigma})^*$. For example,

$$\rho_{\{a,b\}}\left(\boxed{a, c}\boxed{a, b}\boxed{c, d}\right) \ = \ \boxed{a}\boxed{a, b}\boxed{}.$$

We can reduce satisfaction of an MSO-formula φ to φ's *vocabulary*

$$voc(\varphi) \ := \ \{a \in \Theta \mid a \text{ occurs in } \varphi\}$$

using the function $\rho_{voc(\varphi)}$.

Proposition 2 *For all $\Sigma \in Fin(\Theta)$, $s \in (2^{\Sigma})^*$, and MSO-formula φ with $voc(\varphi) \subseteq \Sigma$,*

$$s \models \varphi \iff \rho_{voc(\varphi)}(s) \models \varphi \ .$$

3.2 Subsumption and Superposition

It will often be convenient to put aside the subscript Σ on ρ_{Σ} and work with a non-deterministic generalization of ρ_{Σ} given by componentwise containment $\supseteq$ between equally long strings of sets α_i and β_i, called *subsumption* $\trianglerighteq$

$$\alpha_1 \cdots \alpha_n \trianglerighteq \beta_1 \cdots \beta_m \iff n = m \text{ and } \beta_i \subseteq \alpha_i \text{ for } i \in [n]$$

(Fernando 2004). Note that $s \trianglerighteq \rho_{\Sigma}(s)$ for all $s \in (2^{\Theta})^*$ and $\Sigma \in Fin(\Theta)$. A useful accompaniment to subsumption $\trianglerighteq$ is *superposition* &, which, given two strings of the same length n, returns their componentwise union

$$\alpha_1 \cdots \alpha_n \,\&\, \beta_1 \cdots \beta_n \ := \ (\alpha_1 \cup \beta_1) \cdots (\alpha_n \cup \beta_n)$$

so that for strings s and s' of the same length

$$s \trianglerighteq s' \iff s \& s' = s.$$

To extend superposition and subsumption to languages L and L', we collect the superpositions of strings of the same length from L and L' for

$$L \,\&\, L' \;:=\; \{s\&s' \mid s \in L,\ s' \in L' \text{ and } \mathrm{length}(s) = \mathrm{length}(s')\}$$

and agree that L *subsumes* L' if L is contained in the superposition $L\&L'$

$$L \trianglerighteq L' \iff L \subseteq L\&L'.$$

The intuition is that strings in a language describe possibilities which combine disjunctively so that conflating (as usual) a string s with the language $\{s\}$,

$$s \trianglerighteq L \iff (\exists s' \in L)\; s \trianglerighteq s'.$$

The previously mentioned association of a language $\mathscr{L}(q)$ with an event q such that

$$q \text{ occurs at } (i,j) \text{ within } s \iff s_i^j \in \mathscr{L}(q)$$

(for any string s) can often be put as

$$q \text{ occurs at } (i,j) \text{ within } s \iff s_i^j \trianglerighteq \mathscr{L}_\circ(q)$$

with $\mathscr{L}(q)$ taking the form $\langle\trianglerighteq\rangle\mathscr{L}_\circ(q)$, where for any relation R between strings and any language L, we write $\langle R\rangle L$ for the preimage of L under R

$$\langle R\rangle L \;:=\; \{s \mid (\exists s' \in L)\; sRs'\}.$$

For example, to mark the beginning of q by b_q and its end by e_q, let

$$\mathscr{L}_\circ(q) \;:=\; \boxed{b_q}\boxed{}^* \;\&\; \boxed{}^*\boxed{e_q} \;=\; \boxed{b_q, e_q} + \boxed{b_q}\boxed{}^*\boxed{e_q}.$$

Generalizing from an event q and a particular choice of $\mathscr{L}_\circ(q)$, consider the problem of attaching brackets to a string s at positions i and j such that s_i^j subsumes a particular language M of interest (e.g. $\mathscr{L}_\circ(q)$), assuming that neither [nor] occurs in s. We use the language of brackets

$$\mathscr{L}_{[]} \;:=\; \boxed{[,]} + \boxed{[}\boxed{}^*\boxed{]}$$

to attach a pair of brackets to a language L (of strings s) via the superposition

$$L\& \;\boxed{}^*\mathscr{L}_{[]}\boxed{}^*$$

which we then intersect with the superposition of L with $\boxed{}^*(M\&\mathscr{L}_{[]})\boxed{}^*$

$$L\&\ \Box^*(M\&\mathscr{L}_{[]})\Box^*$$

to ensure the substring bracketed subsumes M, yielding the language

$$L_![M] \ := \ (L\&\ \Box^*\mathscr{L}_{[]}\Box^*) \ \cap \ (L\&\ \Box^*(M\&\mathscr{L}_{[]})\Box^*).$$

To mark any number of occurrences of an M-string in L (and not just one, as in $L_![M]$), we modify the superpositions in $L_![M]$ slightly for

$$L[M] \ := \ L\&(\mathscr{L}_{[]})_* \ \cap \ L\&(M\&\mathscr{L}_{[]})_*$$

where for any language L', the language L'_* consists of any number of occurrences of strings in L' glued together with empty boxes

$$L'_* \ := \ \Box^*(L'\Box^*)^*.$$

To require that at least one occurrence of M be marked, we intersect $L[M]$ with $\langle \mathrm{hf}\rangle\langle\trianglerighteq\rangle\boxed{[}$, where the relation hf (for "*h*as *f*actor") is the inverse of the factor relation

$$s \ \mathrm{hf}\ s' \quad \iff \quad s' = s_i^j \text{ for some } i,\ j.$$

For at least n occurrences of M, intersect $L[M]$ with $\langle \mathrm{hf}\rangle\langle\trianglerighteq\rangle(\boxed{[}\,\Box^*\boxed{]})^n$. But how do we make sure that all occurrences of M in an $s \in L$ are bracketed?

As the bracket pairs in $\mathscr{L}_{[]}$ do not cross (i.e., a left bracket [cannot appear after a [that has not yet been closed by a matching]), there is no hope catching crossing occurrences of M in L by superposition with $\mathscr{L}_{[]}$. But suppose

($\natural$) any two crossing occurrences of M in L are covered by a larger occurrence of M in L.

($\natural$) holds, for example, if for all $a_1 \cdots a_n$ in L and integers i, j, i', j' such that $1 \leq i \leq i' \leq j \leq n$ and $i' \leq j' \leq n$,

$$a_i \cdots a_j \trianglerighteq M \ \text{ and } \ a_{i'} \cdots a_{j'} \trianglerighteq M \ \text{ implies } \ a_i \cdots a_{max(j,j')} \trianglerighteq M$$

— a property familiar in the literature on statives, illustrated by the inference in (5).[1]

(5) Al was asleep from midnight to 3 am. He was also asleep from 2 to 4 am.
$\therefore$ Al was asleep from midnight to 4am.

Under ($\natural$), we can bracket maximal M-occurrences in L as follows. First, we revise $\mathscr{L}_{[]}$ by adding an auxiliary symbol $\bullet$ to fill in boxes between [and]

[1]In terms of temporal traces $\tau(q)$, the idea is that if q and q' are Q-states such that $\tau(q)$ and $\tau(q')$ intersect, then there is a Q-state with trace $\tau(q) \cup \tau(q')$.

$$\mathscr{L}_{[\bullet]} \;:=\; \mathscr{L}_{[]}\& \left(\boxed{} + \boxed{}\boxed{\bullet}^{*}\boxed{}\right) \;=\; \boxed{[,]} + \boxed{[}\boxed{\bullet}^{*}\boxed{]}$$

and turn $L[M]$ into

$$L_{\bullet}[M] \;:=\; L\&(\mathscr{L}_{[\bullet]})_{*} \,\cap\, L\&(M\&\mathscr{L}_{[\bullet]})_{*}.$$

To ensure that every substring of L subsuming M is contained between brackets, we then intersect $L_{\bullet}[M]$ with the set of strings s such that every factor of s that subsumes M also subsumes a factor of $\mathscr{L}_{[\bullet]}$. Next, we provide a finite-state method to form such a language and more.

3.3 Constraints and Compression

For any languages L_1 and L_2, let $L_1 \Rightarrow L_2$ be the set of strings s such that every factor of s subsuming L_1 also subsumes L_2. The counter-examples to the constraint $L_1 \Rightarrow L_2$ form the language

$$\text{cEx}[L \Rightarrow L'] \;:=\; \langle \text{hf}\rangle(\overline{\langle \trianglerighteq \rangle L_2} \cap \langle \trianglerighteq \rangle L_1)$$

of strings with factors that subsume L_1 but not L_2, which we negate for

$$L_1 \Rightarrow L_2 \;=\; \overline{\text{cEx}[L_1 \Rightarrow L_2]}$$

(showing $L_1 \Rightarrow L_2$ is a regular language if L_1 and L_2 are). Now, assuming $(\natural)$ above, we can bracket all maximal M-strings in L by intersecting $L_{\bullet}[M]$ with the constraint

$$M \Rightarrow \text{Marked}$$

where Marked is the set of factors of strings in $\mathscr{L}_{[\bullet]}$

$$\text{Marked} \;:=\; \left(\boxed{[} + \varepsilon\right)\boxed{\bullet}^{*}\left(\boxed{]} + \varepsilon\right).$$

For example, if a is a fluent representing a stative, then for $M = \boxed{a}^{+}$, the brackets in

$$L_{\bullet}[M] \cap \;\; (M \Rightarrow \text{Marked})$$

pick out the intervals of the stative's so-called pofective (Galton 1984. page 81).

Recall from the beginning of this section, the functions $f_t : Q \to \{0, 1, 2\}$ constructed from intervals $\tau(q) \subseteq T$ assigned to events $q \in Q$. This construction generalizes to the situation where $q \in Q$ is replaced by a fluent $a \in \Sigma$, and $\tau(q)$ by a union $\tau(a)$ of $\leq m$ intervals of T, for some fixed positive integer m. We map a point $t \in T$ to a function $g_t : \Sigma \to \{0, 1, \ldots, 2m\}$ where for all $a \in \Sigma$, $g_t(a)$ is chosen to be the smallest integer in $\{0, 1, \ldots, 2m\}$ such that

$$g_t(a) \text{ is odd} \iff t \in \tau(a) \tag{§}$$

and whenever $t' \leq t$, $g_{t'}(a) \leq g_t(a)$. Now suppose Σ is finite. Then so is the function space $\Sigma \to \{0, 1, \ldots, 2m\}$ into which the linearly ordered set T is projected by the mapping $t \mapsto g_t$. And as with the f_t's for finite Q, there is a string $b_1 \cdots b_n$ of functions from Σ to $\{0, 1, \ldots, 2m\}$ such that

$$\{g_t \mid t \in T\} = \{b_i \mid i \in [n]\} \text{ and } b_1 < b_2 < \cdots < b_n$$

where $\leq$ on functions $g, g' : \Sigma \to \{0, 1, \ldots, 2m\}$ is defined componentwise as before

$$g \leq g' \iff (\forall a \in \Sigma)\ g(a) \leq g'(a).$$

The string $b_1 \cdots b_n$ belongs to $(2^{\Sigma'})^*$, where Σ' is the product $\Sigma \times \{0, 1, \ldots, 2m\}$, making it natural to construe a pair $(a, i) \in \Sigma'$ as a fluent. Under the biconditional (§) above, the fluent a is essentially the disjunction $\bigvee\{(a, i) \mid i \in [2m] \text{ and i is odd}\}$. In practice, however, the extra information i that (a, i) provides over a is not always available, and it is useful to leave m and $i \in \{0, 1, \ldots, 2m\}$ unspecified with strings over the alphabet 2^Σ, as opposed to the function space $\Sigma \to \{0, 1, \ldots, 2m\}$.

Behind the reduction of T to $b_1 \cdots b_n$ is "McTaggart's dictum that 'there could be no time if nothing changed'" (Prior 1967, page 85), which we can implement over the alphabet 2^Σ by working with strings $\alpha_1\alpha_2 \cdots \alpha_k \in (2^\Sigma)^+$ that are *stutter-free* in that $\alpha_i \neq \alpha_{i+1}$ for i from 1 to $k - 1$. An equivalent way of characterizing stutter-free strings is through the biconditional

$$s \text{ is stutter-free} \iff s = bc(s)$$

where the *block compression* $bc(s)$ *of* s compresses blocks α^j of $j > 1$ consecutive occurrences in s of the same symbol α to a single α, leaving s otherwise unchanged

$$bc(s) \ := \begin{cases} bc(\alpha s') & \text{if } s = \alpha\alpha s' \\ \alpha\ bc(\beta s') & \text{if } s = \alpha\beta s' \text{ with } \alpha \neq \beta \\ s & \text{otherwise.} \end{cases}$$

For example,

$$bc\left(\boxed{}\boxed{}\boxed{e}\boxed{e}\boxed{e}\boxed{e, y}\boxed{e, y}\boxed{e}\right) = \boxed{}\boxed{e}\boxed{e, y}\boxed{e}.$$

Apart from applying bc, we can make a string s stutter-free by introducing a fresh fluent, say tic, superposing s with

Table 1 A minimal picture of ±durativity and φ-telicity

	−durative	+durative
−telic	Semelfactive $\boxed{\text{ap}(f)}\boxed{\text{ef}(f)}$	Activity $\boxed{\text{ap}(f)}\boxed{\text{ap}(f), \text{ef}(f)}^{+}\boxed{\text{ef}(f)}$
+telic	Achievement $\boxed{\neg\varphi}\boxed{\varphi}$	Accomplishment $\boxed{\neg\varphi, \text{ap}(f)}\boxed{\neg\varphi, \text{ap}(f), \text{ef}(f)}^{+}\boxed{\varphi, \text{ef}(f)}$

$$\left(\boxed{\text{tic}}\boxed{}\right)^{*}\left(\boxed{\text{tic}} + \varepsilon\right)$$

to turn, for instance, $\boxed{a}\boxed{a}\boxed{a}$ into $\boxed{a,\text{tic}}\boxed{a}\boxed{a,\text{tic}}$. Similarly, to extend the string

$$s_\Sigma \;:=\; \boxed{\text{Jan}}\boxed{\text{Feb}}\cdots\boxed{\text{Dec}}$$

of length 12, we add days d1,. . ., d31 to Σ := {Jan,. . .,Dec} for

$$\boxed{\text{Jan,d1}}\boxed{\text{Jan,d2}}\cdots\boxed{\text{Jan,d31}}\boxed{\text{Feb,d1}}\cdots\boxed{\text{Dec,d31}}$$

which ρ_Σ maps to

$$\boxed{\text{Jan}}^{31}\cdots\boxed{\text{Dec}}^{31}$$

and which bc maps back to s_Σ. The crucial point is that stutter-freeness ensures the vocabulary is large enough to express the distinctions of interest.

The choice of vocabulary is key to the Vendler classes described in Dowty (1979) and variants thereof (e.g. Moens et al. 1988). Given a representation of an event q at granularity Σ as a string $str_\Sigma(q) \in (2^\Sigma)^+$, we define q to be

- *Σ-telic* if $str_\Sigma(q) \trianglerighteq \boxed{\neg\varphi}^{+}\boxed{\varphi}$ for some fluent $\varphi \in \Sigma$ (marking the culmination of q)
- *Σ-dynamic* if $\mathit{bc}(str_\Sigma(q))$ has length ≥ 2
- *Σ-durative* if $\mathit{bc}(str_\Sigma(q))$ has length ≥ 3

(Fernando 2015). Apart from transitions $\boxed{\neg\varphi}\boxed{\varphi}$ in which a fluent φ representing a state becomes true, we form transitions $\boxed{\text{ap}(f)}\boxed{\text{ef}(f)}$ recording an effectual application of a force f, with the intuition that

ap(f)	says	"a force f is applied"
ef(f)	says	"a (previous application of) force f was effectual."

The transitions $\boxed{\text{ap}(f)}\boxed{\text{ef}(f)}$ and $\boxed{\neg\varphi}\boxed{\varphi}$ describe semelfactives and achievements, respectively, together forming the non-durative column in Table 1.

Iterating the transitions $\boxed{\text{ap}(f)}\boxed{\text{ef}(f)}$ yields the language

$$\mathscr{L}(f) \;:=\; \boxed{\text{ap}(f)}\boxed{\text{ap}(f), \text{ef}(f)}^{*}\boxed{\text{ef}(f)}$$

which we superpose with $\boxed{\ }\boxed{\ }\boxed{\ }^{+}$ for the language

$$\boxed{\text{ap}(f)}\boxed{\text{ap}(f), \text{ef}(f)}^{+}\boxed{\text{ef}(f)}$$

in Table 1's activity entry (−telic, +durative), and superpose further with $\boxed{\neg\varphi}^{+}\boxed{\varphi}$ for Table 1's accomplishment entry (+telic, +durative). Note that the block compression of the +durative strings in Table 1 have length 3 (as required). Moreover, the four languages in Table 1 can be obtained from $\mathscr{L}(f)$ and $\boxed{\ }\boxed{\ }$ using the three operators

$$\begin{aligned} \mathit{dur}(L) \;&:=\; L \& \;\boxed{\ }\boxed{\ }\boxed{\ }^{+} \\ \mathit{non\text{-}dur}(L) \;&:=\; L - \mathit{dur}(L) \\ \mathit{cul}(L, \varphi) \;&:=\; L \& \;\boxed{\neg\varphi}^{+}\boxed{\varphi} \end{aligned}$$

that pick out the durative, non-durative and φ-telic strings in L, respectively. The notion of force alluded to in $\mathscr{L}(f)$ is linked to fluents φ that persist forward and backward in the absence of the application of a force against or for φ. More precisely, if we let

force(φ) be ap(f) for some force f with ef(f) $:= \varphi$

then

- the constraint

$$\boxed{\ }\boxed{\varphi} \;\Rightarrow\; \left(\boxed{\varphi}\boxed{\ } + \boxed{\text{force}(\varphi)}\boxed{\ }\right)$$

  says φ persists backward unless a force was applied making it true
- the constraint

$$\boxed{\varphi}\boxed{\ } \;\Rightarrow\; \left(\boxed{\ }\boxed{\varphi} + \boxed{\text{force}(\neg\varphi)}\boxed{\ }\right)$$

  says φ persists forward unless opposed by a force, and
- the constraint

$$\boxed{\text{force}(\varphi)}\boxed{\ } \;\Rightarrow\; \left(\boxed{\ }\boxed{\varphi} + \boxed{\text{force}(\neg\varphi)}\boxed{\ }\right)$$

  says an application of a force for φ is effectual unless opposed.

We do *not* assume that for every force f, there is a fluent φ with $\mathrm{ef}(f) := \varphi$ that is subject to the three constraints above. The first two constraints (i.e. inertia) may fail to apply when the change described is incremental; for example, an increase in the degree $deg[\psi]$ associated with a claim ψ

$$\uparrow deg[\psi] \ := \bigvee_{d \in D[\psi]} (d \leq deg[\psi]) \wedge Previous(deg[\psi] < d))$$

over some set $D[\psi]$ of ψ-degrees (such as temperatures, for the claim ψ that "the soup is hot"). It is understood above that *Previous* is the obvious temporal operator which, in the case of $\uparrow deg[\psi]$, unwinds to the language

$$\boxed{}\boxed{\uparrow deg[\psi]} \ \Leftrightarrow \sum_{d \in D[\varphi]} \boxed{deg[\psi] < d}\boxed{d \leq deg[\psi]}$$

where $L \Leftrightarrow L'$ abbreviates the intersection of $L \Rightarrow L'$ and $L' \Rightarrow L$. To keep the alphabet of the language finite, the set $D[\varphi]$ must be assumed finite, and indefinitely refinable, as any finite set chosen for $D[\varphi]$ can be expanded (with a larger vocabulary $\Sigma \in Fin(\Theta)$).

4 Conclusion

Attribute value structures are formulated as states of deterministic finite automata serving as models of Hennessy-Milner logic in Sect. 2. States are located in strings encoding timelines in Sect. 3, based on a mapping of a state q to a language $\mathscr{L}(q)$ and a satisfaction relation $\models$ such that for any string s and string positions i, j,

$$s, i, j \models \mathrm{occurs}(q) \iff s_i^j \in \mathscr{L}(q) \qquad (\star)$$

(where $(a_1 \cdots a_n)_i^j := a_i \cdots a_j$). The language $\mathscr{L}(q)$ typically has the form of a set $\langle \trianglerighteq \rangle \mathscr{L}_\circ(q)$ of strings that subsume some language $\mathscr{L}_\circ(q) \subseteq (2^\Sigma)^+$, in which case two notions of part, $s \mapsto s_i^j$ and $s_i^j \trianglerighteq s'$, take s in $(\star)$ to an element $s' \in \mathscr{L}_\circ(q)$

$$\begin{aligned} s, i, j \models \mathrm{occurs}(q) &\iff (\exists s' \in \mathscr{L}_\circ(q))\ s_i^j \trianglerighteq s' \\ &\iff s \in \langle \cdot_i^j \rangle \langle \trianglerighteq \rangle \mathscr{L}_\circ(q). \end{aligned}$$

By contrast, truthmaking in Fine (2015) is based on a single part relation $\sqsubseteq$ for a *state space* $\langle S, \sqsubseteq \rangle$, on top of which a subset $S^\diamond \subseteq S$ of *possible* states is introduced for a *modalized state space* $\langle S, S^\diamond, \sqsubseteq \rangle$. Of course, just as $S^\diamond$ might be constructed by intersecting any number of constraints, so too might $\sqsubseteq$ be composed from any number of part relations — for example (switching to the converse $\sqsupseteq$), the composition $\cdot_i^j ; \trianglerighteq$ of $\cdot_i^j$ followed by $\trianglerighteq$, under which the preimage of a language L can be calculated through two successive preimages

$$\langle \cdot_i^j ; \trianglerighteq \rangle L \;=\; \langle \cdot_i^j \rangle \langle \trianglerighteq \rangle L$$

figuring in ($\star$) above. The question then, however, is whether it is helpful to keep the various constraints and part relations distinct, or whether undifferentiated lumps $S^{\Diamond}$ and $\sqsubseteq$ suffice. The present work proceeds from the hypothesis that the former is the case.

A simple sentence such as (6) may well be made true by an event of Brutus stabbing Caesar, but noteworthy too is the past tense in stab*bed*.

(6) Brutus stabbed Caesar.

The approach above analyzes tense by grounding an attribute-value account of events in a timeline, encoded by a string. In particular, a Hennessy-Milner formula ψ can be interpreted relative to a string s and string positions i, j as the disjunction

$$\bigvee \{\text{occurs}(q) \mid q \in \mathit{domain}(\mathscr{L}) \text{ and } q \models^{HM} \psi\}$$

over states q satisfying ψ (in the sense of Hennessy-Milner), setting

$$s, i, j \models \psi \iff s_i^j \in \bigcup \{\mathscr{L}(q) \mid q \in \mathit{domain}(\mathscr{L}) \text{ and } q \models^{HM} \psi\}$$

and treating a state q as part of a string s according to $\mathscr{L}$

$$q <_{\mathscr{L}} s \iff q \in \mathit{domain}(\mathscr{L}) \text{ and } s \in \mathscr{L}(q)$$

so that

$$s, i, j \models \psi \iff (\exists q <_{\mathscr{L}} s_i^j)\; q \models^{HM} \psi.$$

The reference interval (i, j) is manipulable through modal operators

$$s, i, j \models \langle R \rangle \varphi \iff (\exists k, l)\; (i, j)[\![R]\!](k, l) \text{ and } s, k, l \models \varphi$$

with, for example, (i, j) shifted back, assuming

$$(i, j)[\![R]\!](k, l) \iff 1 \leq k \leq l \leq i \leq j.$$

Each of the parameters i, j (alongside s, to the left of $\models$) is akin to a free variable in an MSO-formula, and can be absorbed into a string by attaching fresh fluents [and] to the ith and jth positions of s for the string

$$s_{i,j} \;:=\; s \& \boxed{}^{i-1} \boxed{[}\boxed{}^{*} \& \boxed{}^{j-1} \boxed{]}\boxed{}^{*}.$$

The fluents [and] are fresh insofar as they do not already occur in s. While we might require [and] to have unique occurrences in $s_{i,j}$, it is not obvious that whenever $s, i, j \models \varphi$, a truthmaker for φ must be part of the factor s_i^j of s. Indeed, if (i, j) marks the speech time, then reports in the past tense describe events prior to (i, j).

The satisfaction relation $\models$ above is not exact in the way (Fine 2015) understands truthmaking to be. That is, the string $s_{i,j}$ need not be wholly relevant to a formula φ that is $\models$-satisfied by s, i, j (Fernando 2015b). Of course, one may well challenge the presupposition here of a string s. A more common starting point is a linearly ordered set T of temporal moments (such as the real line; e.g., Kamp and Reyle 1993). For any such T, Sect. 3 shows how to extract a string of

- functions from a set Q of events to $\{0, 1, 2\}$, based on projections of events $q \in Q$ as intervals $\tau(q)$ of T (treating q's as particulars), or
- functions from a set Σ of fluents to $\{0, 1, \ldots, 2m\}$, from projecting fluents $a \in \Sigma$ to unions $\tau(a)$ of $\leq m$ intervals (treating a's as finitely repeatable universals).

Both constructions depend on the choice of a finite set; in the former case, Q; in the latter, Σ. The aforementioned languages $\mathscr{L}(q) \subseteq (2^{\Sigma})^*$ bring these constructions together (through the alphabet 2^{Σ} of $\mathscr{L}(q)$). We come back to a formula φ and the language

$$\mathscr{L}(\varphi) \ := \ \{s_{i,j} \mid s, i, j \models \varphi\}$$

it defines. $\mathscr{L}(\varphi)$ is regular precisely if φ can be formulated in MSO. Such formulability (in principle) need not mean restricting the practice of finite-state methods to MSO (any more than say, to regular expressions). For declarative (as opposed to procedural) perspectives, there is an abstract notion of an *institution* due to Goguen et al. 1992), covering MSO and Hennessy-Milner logics on deterministic finite automata (Fernando 2015a). Propositions 1 and 2 (in §§2.2 and 3.1 above) establish the *satisfaction condition* characteristic of institutions, highlighting certain projections, $q \mapsto (q \restriction A)_s$ and $\rho_{voc(\varphi)}$, that return parts dependent on the bounded granularity chosen, A and $voc(\varphi)$. For this reason, Propositions 1 and 2 have been singled out from the other assertions made above. In line with the bias towards logical pluralism and heterogeneity in Goguen and Burstall (1992), the intention is *not* to downplay other notions of part or of truthmaking. The point rather is to illustrate the diversity of such notions, and the usefulness in lifting a notion $T(\varphi)$ of truthmaking to a set C of constraints and a part relation $\sqsupseteq$ (analogous to a modalized state space in Fine (2015) for a language

$$\mathscr{L}(\varphi) \ = \ C \cap \langle \sqsupseteq \rangle T(\varphi)$$

that is regular, provided $T(\varphi)$ and C are regular and $\sqsupseteq$ is computed by a finite-state transducer.

References

Beaver, D., & Condoravdi, C. (2007). On the logic of verbal modification. In M. Aloni, P. Dekker & F. Roelofsen (Eds.), *Proceedings of the 16th Amsterdam Colloquium* (pp. 3–9). Amsterdam.
Blackburn, P. (1993). Modal logic and attribute value structures. In M. de Rijke (Ed.), *Diamonds and Defaults* (pp. 19–65). Netherland: Kluwer.
Brzozowski, J. A. (1964). Derivatives of regular expressions. *Journal of the ACM*, *11*, 481–494.
Carnap, R. (1947). *Meaning and Necessity* (2nd ed.). Chicago: University of Chicago Press.
Champollion, Lucas. (2015). The interaction of compositional semantics and event semantics. *Linguistics and Philosophy*, *38*, 31–66.
Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation*, *15*(2), 99–112.
Cooper, R. (2016). Frames as records. In A. Foret et al. (Ed.), *Formal Grammar* (Vol. 9804). LNCS Heidelberg: Springer.
Cooper, R., & Ginzburg, J. (2015). TTR for natural language semantics. In S. Lappin & C. Fox, et al. (Eds.), *Handbook of Contemporary Semantic Theory* (pp. 375–407), (2nd ed.). New Jersey: Wiley-Blackwell.
Davidson, D. (1967). The logical form of action sentences. In N. Rescher (Ed.), *The Logic of Decision and Action* (pp. 81–95). Pennsylvania: University of Pittsburgh Press.
Dowty, D. R. (1979). *Word Meaning and Montague Grammar*. Boston: Reidel.
Fernando, Tim. (2004). A finite-state approach to events in natural language semantics. *Journal of Logic and Computation*, *14*(1), 79–92.
Fernando, T. (2015). The semantics of tense and aspect: A finite-state perspective. In S. Lappin & C. Fox. (Eds.), *Handbook of Contemporary Semantic Theory* (pp 203–236), (2nd ed.). New Jersey: Wiley-Blackwell.
Fernando, T. (2015a). Two perspectives on change and institutions. Formal Ontologies for AI, http://ceur-ws.org/Vol-1517/JOWO-15_FOfAI_paper_2.pdf.
Fernando, T. (2015b). Negation and events as truthmakers. 20th Amsterdam Colloquium, pp 109–118, http://semanticsarchive.net/Archive/mVkOTk2N/AC2015-proceedings.pdf.
Fernando, T. (2016). Types from frames as finite automata. In A. In Foret, et al. (Eds.), *Formal Grammar* (Vol. 9804), LNCS (pp. 19–40). Heidelberg: Springer.
Fillmore, C. J. (1982). Frame semantics. In *Linguistics in the Morning Calm* (pp 111–137) Seoul: (Hanshin Publishing Co.)
Fine, K. (2015). Truthmaker semantics. Draft chapter for the Blackwell Philosophy of Language Handbook. New Jersey: Wiley-Blackwell.
Galton, A. (1984). *The Logic of Aspect: An Axiomatic Approach*. Oxford: Clarendon Press.
Goguen, Joseph A., & Burstall, Rod M. (1992). Institutions: Abstract model theory for specification and programming. *Journal ACM*, *39*, 95–146.
Hennessy, M., & Milner, R. (1985). Algebraic laws for non-determinism and concurrency. *Journal ACM*, *32*(1): 137–161.
Hopcroft, J., & Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation*. Boston: Addison-Wesley.
Kamp, H., & Reyle, U. (1993). *From Discourse to Logic*. Netherland: Kluwer.
Libkin, L. (2004). *Elements of Finite Model Theory*. Heidelberg: Springer.
Luo, Z. (2012). Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, *35*(6), 491–513.
Moens, M., & Steedman, M. (1988). Temporal ontology and temporal reference. *Computational Linguistics*, *14*(2), 15–28.
Montague, R. (1974). *Formal Philosophy*. London: Yale University Press.
Mulligan, K., Simons, P., & Smith, B. (1984). Truth-Makers. *Philosophy and Phenomenological Research*, *44*(3), 287–321.
Petersen, W. (2007). Representation of concepts as frames. In *Complex Cognition and Qualitative Science*. The Baltic International (pp. 151– 170). Riga: University of Latvia.

Prior, Arthur N. (1967). *Past, Present and Future*. Oxford: Clarendon Press.
Ranta, A. (1994). *Type-Theoretical Grammar*. New York: Oxford University Press.
Smith, B. (1999). Truthmaker realism. *Australasian Journal of Philosophy*, *77*(3), 274–291.