3

Events from temporal logic to regular languages with branching

TIM FERNANDO

Abstract

Events in natural language semantics, conceived as strings of observations, are extracted from formulas of linear temporal logic, and collected in regular languages. Infinite strings of sets of atomic formulas (fully specifying truth) are truncated and partialized, in line with the bounded temporal extent and descriptive content of events. Branching from that line, counterfactual events are analyzed as b(ranching)-strings accepted by finite b-automata. These structures are compared and contrasted to those of Computational Tree Logic.

Keywords EVENTS, TEMPORAL LOGIC, REGULAR LANGUAGES, BRANCHING

3.1 Introduction

Priorean temporal logics have attracted considerable attention in efforts to verify computational systems (e.g. Emerson (1992), Clarke et al. (1999)). Has that attention been matched by the formal natural language semantics community? Perhaps in the past (e.g. Thomason (1984)). But if in recent years that popularity has waned, some of the blame must be put down to a reluctance to mix events with temporal logic.[1] Freely appealing to worlds and times, the very influential book Dowty (1979) refers to events only informally. Subsequent works such as Parsons (1990), Kamp and Reyle (1993) and Asher and Las-

[1] That said, it is clear from papers such as Blackburn et al. (1996), Condoravdi (2002) and Bennett and Galton (2004) that not everyone shies away from such a mix.

FG-MoL 2005.
James Rogers (ed.).

carides (2003), however, attest to natural language semantics' enduring interest in events as full-fledged theoretical entities. It is against this background that the present work explores the possibility of extracting events from temporal logic formulas, the truth of which they witness. We start in section 2 with the simple case of propositional linear temporal logic given by discrete future operators — LTL, for short. The infinite timelines suitable for analyzing non-terminating reactive systems (in Emerson (1992) and Clarke et al. (1999)) are noted to be at odds with the bounded temporal extent events have (according say, to Reichenbach or Vendler).[2] Similarly, full specification of atomic formulas true at a moment is incompatible with the partial descriptive content of events relative to any fixed moment. Accordingly, the infinite strings of sets of atomic formulas determined by timeline-valuation pairs are truncated and filled with non-atomic formulas, allowing for lazy evaluation. Events are conceptualized as strings of sets of formulas, and an event-type formulated (following Fernando (2004a)) as a set of such strings that may be accepted by a finite automaton — that is, a regular language.

The conception of an event-type as a regular language is extended in section 3 to branching time, motivated to no small degree by sentences such as

(12) Pat stopped the car before it hit the tree.

A natural reading of (12) diverges from the *before* operator in Emerson (1992) (§3.2.2), carrying (as noted in Heinämäki (1972) and Beaver and Condoravdi (2003)) the implication

(13) The car did not hit the tree, but it may well have.

We must (amongst other things) be careful to interpret the words "may well have" in (13) over possibilities ruled out by the first clause in (13). With this in mind, we augment a finite automaton with a binary relation on states to form a finite b-automaton, providing an alternative to the existential operator E in CTL* (e.g. Emerson (1992),§4.2). Whereas finite automata accept strings, finite b-automata accept b(ranching)-strings, constituting regular b-languages.

3.2 Portraying LTL formulas by regular languages

Following for the most part the notation of Emerson (1992) and Clarke et al. (1999), we build the formulas of LTL from a set P of *atomic propositions* and interpret these relative to a set S of *states*, a *timeline*

[2] Henceforth, events in this paper are to be understood from the perspective of natural language semantics, as opposed to programming language theories.

$x : \mathbb{N} \to S$ giving an infinite sequence $x(0), x(1), \ldots$ of states, and a *valuation* $\mathfrak{l} : S \to 2^P$ specifying the set $\mathfrak{l}(s) \subseteq P$ of atomic propositions true at $s \in S$. We lift the notion $\mathfrak{l}$ of truth at states to timelines x

$$x \models_{\mathfrak{l}} p \quad \text{iff} \quad p \in \mathfrak{l}(x(0)) \qquad\qquad \text{for } p \in P$$

and formulas φ, ψ generated from conjunction $\wedge$, disjunction $\vee$

$$\begin{aligned} x \models_{\mathfrak{l}} \varphi \wedge \psi \quad &\text{iff} \quad x \models_{\mathfrak{l}} \varphi \text{ and } x \models_{\mathfrak{l}} \psi \\ x \models_{\mathfrak{l}} \varphi \vee \psi \quad &\text{iff} \quad x \models_{\mathfrak{l}} \varphi \text{ or } x \models_{\mathfrak{l}} \psi \end{aligned}$$

and three temporal operators that are interpreted by defining for every timeline x and $n \in \mathbb{N}$ the timeline $x^n : \mathbb{N} \to S$ mapping $m \in \mathbb{N}$ to $x(n+m)$. We have *next* X, *until* U and *release* R

$$\begin{aligned} x \models_{\mathfrak{l}} \mathsf{X}\varphi \quad &\text{iff} \quad x^1 \models_{\mathfrak{l}} \varphi \\ x \models_{\mathfrak{l}} \varphi \mathsf{U} \psi \quad &\text{iff} \quad (\exists n \geq 0)\ (x^n \models_{\mathfrak{l}} \psi \text{ and } (\forall m < n)\ x^m \models_{\mathfrak{l}} \varphi) \\ x \models_{\mathfrak{l}} \varphi \mathsf{R} \psi \quad &\text{iff} \quad (\forall n \geq 0)\ (x^n \models_{\mathfrak{l}} \psi \text{ or } (\exists m < n)\ x^m \models_{\mathfrak{l}} \varphi)\ . \end{aligned}$$

Negation is defined through a map $\overline{\cdot} : P \to P$ on atomic propositions with $\overline{\overline{p}} = p$ (doubling P, if necessary, to $P \times \{+,-\}$ with $\overline{(p,+)} = (p,-)$ and $\overline{(p,-)} = (p,+)$), and the dual (De Morgan) pairs $(\wedge, \vee)$, (U, R), (X, X), where

$$\overline{\varphi \wedge \psi} = \overline{\varphi} \vee \overline{\psi} \qquad\qquad \overline{\varphi \vee \psi} = \overline{\varphi} \wedge \overline{\psi}$$

etc. Henceforth, we require of a valuation $\mathfrak{l}$ that for all $s \in S$ and $p \in P$,

$$p \in \mathfrak{l}(s) \quad \text{iff} \quad \overline{p} \notin \mathfrak{l}(s).$$

We write Φ for the set of LTL formulas, with a designated tautology $\top$ and contradiction $\bot = \overline{\top}$, setting $x \models_{\mathfrak{l}} \top/\bot$ for all/no $x, \mathfrak{l}$.

Next, we give a few instances of a language L over the alphabet 2^Φ portraying a formula $\varphi \in \Phi$, which we presently systematize:

$$\begin{aligned} \boxed{p,q} + \boxed{r} \quad &\text{portrays} \quad (p \wedge q) \vee r \\ \boxed{p}\boxed{q,r} \quad &\text{portrays} \quad p \wedge \mathsf{X}(q \wedge r) \\ \boxed{p}^*\boxed{q} \quad &\text{portrays} \quad p\mathsf{U}q\ . \end{aligned}$$

We adopt the notation of regular languages (with non-deterministic choice $+$, Kleene star $\cdot^*$, etc) and enclose a set of formulas by a box rather than by curly braces $\{\cdot\}$ when that set is understood as a symbol. We extend $\models_{\mathfrak{l}}$ to strings $\mathsf{s} = \alpha_1 \cdots \alpha_n \in (2^\Phi)^*$ conjunctively

$$x \models_{\mathfrak{l}} \alpha_1 \cdots \alpha_n \quad \text{iff} \quad \text{whenever } 1 \leq i \leq n \text{ and } \psi \in \alpha_i,\ x^{i-1} \models_{\mathfrak{l}} \psi\ .$$

A language $L \subseteq (2^\Phi)^*$ is then defined to *portray* a formula $\varphi \in \Phi$ if L provides witnesses for the truth of φ in the sense that for all timelines

x and valuations $\mathfrak{l}$,

$$x \models_{\mathfrak{l}} \varphi \quad \text{iff} \quad (\exists \mathsf{s} \in L)\ x \models_{\mathfrak{l}} \mathsf{s}\ .$$

Notice that by allowing the symbols in L to be subsets of Φ and not just of P, we ensure that for every $\varphi \in \Phi$, there is a regular language portraying φ. Take $\boxed{\varphi}$. But for (say) $\varphi = p\mathsf{U}q$, $\boxed{p}^*\boxed{q}$ is far better than $\boxed{p\mathsf{U}q}$ at bringing out how $p\mathsf{U}q$ may stretch over more than one moment. And if we are to view languages as event-types (i.e. sets of events), then it is noteworthy that $\boxed{p\mathsf{U}q}$ has only one event/string (and a rather poorly drawn out one at that) whereas $\boxed{p}^*\boxed{q}$ has infinitely many. Each of the strings $\boxed{p}^n\boxed{q}$ is a more plausible sequence of snapshots than $\boxed{p\mathsf{U}q}$ given that the truth of p and q depends on states (in isolation), unlike the compound $p\mathsf{U}q$.

To pick out portrayals L of φ of the sort given by $\boxed{p}^*\boxed{q}$, let us restrict the alphabet of L a bit, defining the *basis of* φ to be the set $B(\varphi)$ of formulas

$$\begin{array}{ll} B(p) = \{p\} & B(\varphi \bullet \psi) = B(\varphi) \cup B(\psi) \text{ for } \bullet \in \{\wedge, \vee, \mathsf{U}\} \\ B(\mathsf{X}\varphi) = B(\varphi) & B(\varphi \mathsf{R} \psi) = B(\varphi) \cup B(\psi) \cup \{\bot \mathsf{R} \psi\} \end{array}$$

and $B(\varphi) = \emptyset$ for $\varphi \in \{\top, \bot\}$. We set $B_0(\varphi) = B(\varphi) \cap P$, and note that for φ containing *no* occurrences of R, $B(\varphi) = B_0(\varphi)$. Membership of $\bot\mathsf{R}\psi$ in $B(\varphi\mathsf{R}\psi)$ hints at a certain irreducibility of $\varphi\mathsf{R}\psi$ over finite strings. We have

Theorem 1. *For all $\varphi \in \Phi$ and $n > 0$, there is a language $L \subseteq (2^{B_0(\varphi)})^n (2^{B(\varphi)})^*$ that portrays φ.*

Let us define for every $\varphi \in \Phi$ a language $\mathcal{L}(\varphi)$ meeting the specification of Theorem 1. The cases of $p \in P, \vee, \mathsf{X}, \top$ and $\bot$ are easy:

$$\begin{array}{ll} \mathcal{L}(p) = \boxed{p}\,\Box^{n-1}\Box^* & \mathcal{L}(\mathsf{X}\varphi) = \Box\mathcal{L}(\varphi) \\ \mathcal{L}(\varphi \vee \psi) = \mathcal{L}(\varphi) + \mathcal{L}(\psi) & \mathcal{L}(\top) = \Box^n\Box^* \end{array}$$

and $\mathcal{L}(\bot) = \emptyset$. For conjunction, take

$$\mathcal{L}(\varphi \wedge \psi) \quad = \quad \mathcal{L}(\varphi)\ \&\ \mathcal{L}(\psi)$$

where the *superposition* $L\&L'$ *of* languages L, L' over the alphabet 2^Φ is the componentwise union of strings in L and L' of the same length

$$L\&L' = \bigcup_{n \geq 0} \{(\alpha_1 \cup \alpha_1') \cdots (\alpha_n \cup \alpha_n') \mid \alpha_1 \cdots \alpha_n \in L \text{ and } \alpha_1' \cdots \alpha_n' \in L'\}$$

(Fernando, 2004a). As for U, we can*not* generalize the portrayal of $p\mathsf{U}q$ by $\boxed{p}^*\boxed{q}$ by equating $\mathcal{L}(\varphi\mathsf{U}\psi)$ with $\mathcal{L}(\varphi)^*\mathcal{L}(\psi)$. (Take $\varphi = \mathsf{X}p$

and $\psi = p$.) Instead, let us define (simultaneously with $\mathcal{L}$) languages $\mathcal{L}_0(\varphi), \mathcal{L}_1(\varphi), \ldots$ that portray $\varphi^0 = \top, \varphi^1 = \varphi \wedge X\varphi^0, \ldots$

$$\begin{aligned} \mathcal{L}_0(\varphi) &= \Box^* \\ \mathcal{L}_{k+1}(\varphi) &= \mathcal{L}(\varphi) \ \& \ \Box\mathcal{L}_k(\varphi) \\ &= \mathcal{L}(\varphi^{k+1}) \ \text{ where } \varphi^{k+1} = \varphi \wedge X\varphi^k \end{aligned}$$

and set

$$\mathcal{L}(\varphi \mathsf{U} \psi) = \sum_{k \geq 0} (\mathcal{L}_k(\varphi) \ \& \ \Box^k \mathcal{L}(\psi))$$

where $\sum$ is $\bigcup$, just as $+$ is $\cup$. For R, the idea is that for a fixed $n \geq 0$,

$$\boxed{q}^* \boxed{p,q} \ + \ \boxed{q}^n \boxed{\bot \mathsf{R} q} \quad \text{portrays} \quad p\mathsf{R}q$$

which generalizes to

$$\mathcal{L}(\varphi \mathsf{R} \psi) = \sum_{k \geq 0} (\mathcal{L}_k(\psi) \ \& \ \Box^k(\mathcal{L}(\varphi) \& \mathcal{L}(\psi))) \ + \ (\mathcal{L}_n(\psi) \ \& \ \Box^n \boxed{\bot \mathsf{R} \psi}) \ .$$

This completes the definition of $\mathcal{L}$.

Are all the languages $\mathcal{L}(\varphi)$ regular? No (as the sums $\sum$ for U and R are infinite). For $p, q, r \in P$, let

$$\hat{\varphi} = (p \wedge (\top \mathsf{U} q)) \ \mathsf{U} \ r$$

and note (after a moment's reflection) that every string in $\mathcal{L}(\hat{\varphi})$ which happens to be in $\boxed{p}^+ \boxed{r} \boxed{q}^+$ must have no more q's than p's (as a q in a string from $\mathcal{L}(\hat{\varphi})$ pairs up with a p). That is,

$$\mathcal{L}(\hat{\varphi}) \cap \boxed{p}^+ \boxed{r} \boxed{q}^+ = \sum_{i \geq 1} \boxed{p}^i \boxed{r} \sum_{1 \leq j \leq i} \boxed{q}^j$$

which is non-regular (by a pumping argument). Thus, $\mathcal{L}(\hat{\varphi})$ cannot be regular.

To form only regular languages, we must do something about $\mathcal{L}(\varphi \mathsf{U} \psi)$. Let us modify $\mathcal{L}$ to $\hat{\mathcal{L}}$, retaining the portrayals of formulas with*out* U or R

$$\begin{aligned} \hat{\mathcal{L}}(\top) &= \Box^n \Box^* & \hat{\mathcal{L}}(\varphi \vee \psi) &= \hat{\mathcal{L}}(\varphi) + \hat{\mathcal{L}}(\psi) \\ \hat{\mathcal{L}}(\bot) &= \emptyset & \hat{\mathcal{L}}(\mathsf{X}\varphi) &= \Box \hat{\mathcal{L}}(\varphi) \\ \hat{\mathcal{L}}(p) &= \boxed{p} \Box^{n-1} \Box^* & \hat{\mathcal{L}}(\varphi \wedge \psi) &= \hat{\mathcal{L}}(\varphi) \ \& \ \hat{\mathcal{L}}(\psi) \end{aligned}$$

($L \& L'$ is regular if L and L' are (Fernando, 2004a)). The key to U is lazy evaluation; instead of unwinding $p\mathsf{U}q$ fully to $\boxed{p}^* \boxed{q}$, we make do with

$$\boxed{q} + \boxed{p}\boxed{q} + \boxed{p}\boxed{p}\boxed{q} + \cdots + \boxed{p}^{n-1} \boxed{q} + \boxed{p}^n \boxed{p \mathsf{U} q} \quad \text{portrays} \quad p\mathsf{U}q$$

from which we get

$$\hat{\mathcal{L}}(\varphi \mathsf{U} \psi) \quad = \quad \sum_{k<n} (\hat{\mathcal{L}}_k(\varphi) \ \& \ \Box^k \hat{\mathcal{L}}(\psi)) \ + \ (\hat{\mathcal{L}}_n(\varphi) \ \& \ \Box^n \boxed{\varphi \mathsf{U} \psi})$$

where just as with $\mathcal{L}_k$, $\hat{\mathcal{L}}_0(\varphi) = \Box^*$ and $\hat{\mathcal{L}}_{k+1}(\varphi) = \hat{\mathcal{L}}(\varphi) \ \& \ \Box\hat{\mathcal{L}}_k(\varphi)$. Similarly,

$$\sum_{k<n} \boxed{q}^k \boxed{p, q} + \boxed{q}^n \boxed{p\mathsf{R}q} \qquad \text{portrays} \qquad p\mathsf{R}q$$

and so

$$\hat{\mathcal{L}}(\varphi \mathsf{R} \psi) \quad = \quad \sum_{k<n} (\hat{\mathcal{L}}_k(\psi) \ \& \ \Box^k (\hat{\mathcal{L}}(\varphi) \& \hat{\mathcal{L}}(\psi))) \ + \ (\hat{\mathcal{L}}_n(\psi) \ \& \ \Box^n \boxed{\varphi \mathsf{R} \psi}) \ .$$

So much for the definition of $\hat{\mathcal{L}}$. To formulate an analogue to Theorem 1, let us replace $B(\varphi)$ by $C(\varphi)$, where

$$\begin{array}{ll} C(p) = \{p\} & C(\varphi \bullet \psi) = C(\varphi) \cup C(\psi) \text{ for } \bullet \in \{\wedge, \vee\} \\ C(\mathsf{X}\varphi) = C(\varphi) & C(\varphi) = \emptyset \text{ for } \varphi \in \{\top, \bot\} \end{array}$$

and $C(\varphi \bullet \psi) = C(\varphi) \cup C(\psi) \cup \{\varphi \bullet \psi\}$ for $\bullet \in \{\mathsf{U}, \mathsf{R}\}$. Let $C_0(\varphi) = C(\varphi) \cap P$.

Theorem 2. *For all $\varphi \in \Phi$ and $n > 0$, there is a language $L \subseteq (2^{C_0(\varphi)})^n (2^{C(\varphi)})^*$ that portrays φ and is regular.*

Neither the language $\hat{\mathcal{L}}(\varphi)$ behind Theorem 2 nor the language $\mathcal{L}(\varphi)$ behind Theorem 1 necessarily offers the most obvious portrayal of φ. For example, $\Box$ portrays every tautology (whether or not that tautology contains R or U), $\emptyset$ portrays every contradiction, and the regular language

$$\hat{L} \quad = \quad \boxed{r} \ + \ \boxed{p}^* (\boxed{p, q}\boxed{r} \ + \ \boxed{p}\boxed{r, q} \ + \ \boxed{p}\boxed{r}\Box^*\boxed{q})$$

portrays the formula $\hat{\varphi} = (p \wedge (\top \mathsf{U} q)) \ \mathsf{U} r$, which (as we noted above) $\mathcal{L}$ maps to a non-regular language.[3] Each of $\hat{L}$, $\mathcal{L}(\hat{\varphi})$ and $\hat{\mathcal{L}}(\hat{\varphi})$ have very different temporal extents, and temporal extent is crucial in event semantics. In particular, we had better not confuse $\boxed{\varphi}^+$ with $\boxed{\varphi}$ as event-types, even though both portray φ. For the record, let us state an obvious but important fact about portrayal, agreeing (as usual) that $\varphi, \psi \in \Phi$ are *logically equivalent* if the same timeline-valuation pairs satisfy them.

---

[3]Observe that $\hat{L}\Box^* = \mathcal{L}(r \vee (p\mathsf{U}(p \wedge \hat{\psi})))$ for $\hat{\psi} = (q \wedge \mathsf{X}r) \vee \mathsf{X}(r \wedge q) \vee \mathsf{X}(r \wedge \mathsf{X}(\top \mathsf{U} q))$. Excess $\Box$'s at the end of a string in a language L can be stripped off by intersecting L with $\Box + (2^\Phi)^*(2^\Phi - \{\Box\})$.

Proposition 3. *If $\varphi, \psi \in \Phi$ are logically equivalent, then they are portrayed by the same languages. Conversely, if there is a language that portrays both φ and ψ, then φ and ψ are logically equivalent.*

Clearly, event-types as languages are much finer grained notions than formulas in Φ under logical equivalence.[4] But if this is all we want to say, then surely Theorems 1 and 2 are overkill, are they not?

Part of the interest in Theorems 1 and 2 lies in the incremental construction of a timeline-valuation pair $x, \mathfrak{l}$ satisfying a formula φ, which can (for the purpose of $\models$) be reduced to the infinite sequence

$$\mathfrak{l}(x(0)), \mathfrak{l}(x(1)), \mathfrak{l}(x(2)), \ldots$$

Readers familiar with tableaux for LTL (e.g. Clarke et al. (1999)) will have no doubt noticed that progressively larger initial segments of such sequences are given by strings in $\mathcal{L}(\varphi)$ and $\hat{\mathcal{L}}(\varphi)$ that grow as n approaches ∞.[5] (To keep the notation simple, we have decorated neither $\mathcal{L}$ nor $\hat{\mathcal{L}}$ with n; we pay a price for that now.) The only difference between $\mathcal{L}(\varphi)$ and $\hat{\mathcal{L}}(\varphi)$ is the care we exercised in the latter to produce increments that some finite automata can accept. Furthermore, we have refrained from mentioning automata on infinite strings (again, see, for instance, Clarke et al. (1999)) because we want a string that is interpretable as an event, whose time E may precede a *speech time* S (for the past tense, following Reichenbach). If we are to put S and E on the same timeline (with length $\mathbb{N}$), then the string with time E had better be finite so that S can come after E.

In view of the irreducibility of the release operator R over finite strings (accounting for the failure in Theorem 1 of the inclusion $B(\varphi) \subseteq P$), the question arises: does event semantics have any use for R? But why should it *not*? Although an event may be bounded, its effects need not. Consider (12), repeated below.

(12) Pat stopped the car before it hit the tree.

An effect of Pat stopping the car is that car be stationary. We can assume the car remains stationary unless some force puts it in motion, which is, in turn, a precondition for the car hitting the tree. For this reason, we may conclude from (12) that in the absence of any intervening forces, the car did not hit the tree. To formalize such reasoning, let us suppose that for certain formulas φ, we could build a formula $\mathsf{F}\varphi$ saying intuitively that a force is applied on φ. Then the formula

[4] Borrowing terminology from Schubert (2000), event-types collect *characterized* situations, whereas logical equivalence is given by a wider *supports* relation.

[5] The entailments at stake here (definable from &) are related in Fernando and Nairn (2005) to restrictions and replacements in Beesley and Karttunen (2003).

$(\mathsf{F}\varphi)\mathsf{R}\varphi$ says:

(†) φ holds and will continue to do so until a force is applied on it.

The notion of inertia implicit in (†) is developed in Fernando (2004b) to analyze (12) and develop ideas about aspect from Reichenbach and Vendler. But instead of explicitly mentioning R, rules of inertial flow are set out, with $\mathsf{F}\varphi$ read as "freeze φ."

3.3 Branching beyond CTL and regular languages

A natural reading of (12) supports (13), repeated below.

(13) The car did not hit the tree, but it may well have.

A first attempt at interpreting the multiple possibilities in (13) is to relativize satisfaction $\models$ to a set X of timelines alongside $x, \mathfrak{l}$, from which to reset x by an existential operator $\hat{\mathsf{E}}$

$$X, x \models_{\mathfrak{l}} \hat{\mathsf{E}}\varphi \quad \text{iff} \quad (\exists x' \in X)\ X, x' \models_{\mathfrak{l}} \varphi\ .$$

Note that whether or not $X, x \models_{\mathfrak{l}} \hat{\mathsf{E}}\varphi$ holds is independent of x. $\hat{\mathsf{E}}$ can be read as *epistemic might* if X is understood as the conversational common ground, subject to update (e.g. Veltman (1996)). This construal brings the portrayal of $\varphi \vee \psi$ by $\boxed{\varphi} + \boxed{\psi}$ in line with an interpretation of disjunction as a list of epistemic possibilities (Zimmermann, 2000). Restricting quantification to timelines x' with the same initial state $x'(0)$ leads to the existential operator E

$$X, x \models_{\mathfrak{l}} \mathsf{E}\varphi \quad \text{iff} \quad (\exists x' \in X)\ x'(0) = x(0) \text{ and } X, x' \models_{\mathfrak{l}} \varphi$$

and to *state formulas* in CTL* (Emerson (1992),§4.2) with x truncated to $x(0)$. E approximates what Condoravdi (2002) calls *metaphysical might* insofar as the equation $x(0) = x'(0)$ captures alternatives under *historical necessity* (e.g. Thomason (1984)) on a set X consisting, for some binary ("successor") relation R on the set S of states, of timelines x such that for all $i \geq 0$, $x(i)\ R\ x(i+1)$.

Can we use $\hat{\mathsf{E}}$ or E to analyze the *may* in (13)? Treating it epistemically as $\hat{\mathsf{E}}$ will not do, if (as is eminently plausible) the first clause of (13) eliminates timelines where the car hits the tree from the common ground. As for E, it fails to provide a notion of counterfactual event or realis distance to mark 'car did not hit tree' as factual, and 'car hit tree' as an unrealized possibility (according to (13)). That such a notion is missing from CTL* is what is called the "disconnection from the present" (Nelken and Francez, 1996) of system specifications.

But already, CTL* presents a problem for our identification of event-types with languages, rather than the automata accepting them. Evaluating $\mathsf{E}\varphi$ according to $\models$ requires more structure than is provided by a set of strings.[6] Accordingly, we add branches to strings, generating the set Σ^b of (non-empty) *b-strings* s over an alphabet Σ

$$\mathsf{s} \quad :== \quad \alpha \mid \mathsf{s}\mathsf{s}' \mid b(\mathsf{s},\mathsf{s}')$$

from symbols $\alpha \in \Sigma$ by binary operations of concatenation and branching b.[7] We will work with alphabets $\Sigma \subseteq 2^\Phi$ consisting of sets of formulas, and will shortly formalize the intuition that $b(\mathsf{s},\mathsf{s}')$ says s' *may* follow s, whereas $\mathsf{s}\mathsf{s}'$ says s' follows s (without qualification). Hence, if s_1 clashes with s_2, then $b(\mathsf{s},\mathsf{s}_1)\mathsf{s}_2$ describes s_1 as a counterfactual continuation of s, which continues instead along s_2. For example,

$$b(\boxed{\text{moving-car}}, \boxed{\text{moving-car}, \overline{\text{contact}}}\boxed{\text{contact}})\ \boxed{\overline{\text{moving-car}}}\boxed{\text{S}}$$

is a b-string depicting the sentence *The car stopped before it hit the tree* (with speech time S), assuming (for simplicity)

$$\boxed{\text{moving-car}}\boxed{\overline{\text{moving-car}}}\boxed{\text{S}} \quad \text{depicts} \quad \textit{the-car-stopped}$$

and

$$\boxed{\text{moving-car}, \overline{\text{contact}}}\boxed{\text{contact}} \quad \text{depicts} \quad \textit{the-car-hit-the-tree}.$$

For a formal treatment of depiction in terms of portrayal, we step up to sets of b-strings (as with ordinary non-branching strings). We turn a finite automaton (over Σ) with set Q of states, set $\rightarrow \subseteq Q \times \Sigma \times Q$ of transitions, initial state $q_0 \in Q$ and set $F \subseteq Q$ of final states into a *finite b-automaton* (over Σ) by adding to the finite automaton a binary relation $\xrightarrow{b} \subseteq Q \times Q$ on Q (intuitively marking realis as well as temporal distance). We define a ternary relation $\Rightarrow \subseteq Q \times \Sigma^b \times Q$ by

$$\begin{array}{rcl}
q \overset{\alpha}{\Rightarrow} q' & \text{iff} & q \overset{\alpha}{\rightarrow} q' \\
q \overset{\mathsf{s}\mathsf{s}'}{\Rightarrow} q' & \text{iff} & (\exists q'' \overset{\mathsf{s}}{\Leftarrow} q)\ q'' \overset{\mathsf{s}'}{\Rightarrow} q' \\
q \overset{b(\mathsf{s},\mathsf{s}')}{\Rightarrow} q' & \text{iff} & q \overset{\mathsf{s}}{\Rightarrow} q' \text{ and } (\exists q_1 \overset{b}{\leftarrow} q')(\exists q_2 \in F)\ q_1 \overset{\mathsf{s}'}{\Rightarrow} q_2
\end{array}$$

and say the finite b-automaton *accepts* a b-string s if for some $q \in F$, $q_0 \overset{\mathsf{s}}{\Rightarrow} q$. An equivalent presentation of b-strings is provided by the notion

[6] For instance, the difference between the regular expressions $\boxed{r}(\boxed{p}+\boxed{q})$ and $\boxed{r}\boxed{p}+\boxed{r}\boxed{q}$ surfaces when $\models$-evaluating the formula $\mathsf{EX}(r \wedge \mathsf{EX}p \wedge \mathsf{EX}q)$ against the respective structures suggested by the regular expressions. Let $S = \{0,1,2,3,4\}$, $\mathfrak{l}(1) = \mathfrak{l}(2) = \{r\}$, $\mathfrak{l}(3) = \{p\}$, $\mathfrak{l}(4) = \{q\}$ and compare $R = \{(0,1),(1,3),(1,4)\}$ with $R' = \{(0,1),(1,3),(0,2),(2,4)\}$.

[7] We may impose the equations

$$\begin{array}{rclcrcl}
(\mathsf{s}\mathsf{s}')\mathsf{s}'' & = & \mathsf{s}(\mathsf{s}'\mathsf{s}'') & \quad & b(b(\mathsf{s},\mathsf{s}'),\mathsf{s}') & = & b(\mathsf{s},\mathsf{s}') \\
b(\mathsf{s}\mathsf{s}',\mathsf{s}'') & = & \mathsf{s}b(\mathsf{s}',\mathsf{s}'') & \quad & b(b(\mathsf{s},\mathsf{s}'),\mathsf{s}'') & = & b(b(\mathsf{s},\mathsf{s}''),\mathsf{s}')
\end{array}$$

which the automata-theoretic notions below (viz $\overset{\mathsf{s}}{\Rightarrow}$) respect.

of a *b-form* σ (over Σ), defined simultaneously with *b-possibilities* A by

$$\begin{array}{rcl} \sigma & :== & (\alpha, A) \mid \sigma\sigma' \\ A & :== & \emptyset \mid \sigma + A \end{array}$$

for $\alpha \in \Sigma$ with constant $\emptyset$. Each b-string s has a b-form $f(\mathsf{s}) = (\alpha_1, A_1) \cdots (\alpha_n, A_n)$ given by

$$\begin{array}{rcl} f(\alpha) & = & (\alpha, \emptyset) \\ f(\mathsf{ss}') & = & f(\mathsf{s})f(\mathsf{s}') \\ f(b(\mathsf{s}, \mathsf{s}')) & = & (\alpha_1, A_1) \cdots (\alpha_{n-1}, A_{n-1})(\alpha_n, f(\mathsf{s}') + A_n) \ . \end{array}$$

We can then view a finite b-automaton as a 2-sorted top-down tree automaton (e.g. Comon et al. (2002)), and acceptability of s as derivability of $q_0\mathsf{s} \to \mathsf{s}$ from the rules

$$\begin{array}{rcll} q(\alpha, A) & \to & (\alpha, q'A) & \text{for } q \xrightarrow{\alpha} q' \in F \\ q((\alpha, A)\sigma) & \to & (\alpha, q'A)q'\sigma & \text{for } q \xrightarrow{\alpha} q' \\ q(\emptyset) & \to & \emptyset & \\ q(\sigma + A) & \to & q'\sigma + qA & \text{for } q \xrightarrow{b} q' \ . \end{array}$$

Next, let us link $\xrightarrow{b}$ to an interpretation of formulas $\mathsf{may}(\varphi)$ relative to the transitions $\to \subseteq Q \times (\Sigma \cup \{b\}) \times Q$ of a finite b-automaton $(Q, \to, q_0, F)$, forming

(a) timelines $x : \mathbb{N} \to (Q \times \Sigma)$ such that for all $n \geq 0$, $q(n) \xrightarrow{\alpha(n)} q(n+1)$ where $x(n) = (q(n), \alpha(n))$ [assuming wlog domain$(\bigcup_\alpha \xrightarrow{\alpha}) = Q$]

(b) the valuation $\mathfrak{l} : (Q \times \Sigma) \to \Sigma$ mapping (q, α) to α, and

(c) a binary relation $\twoheadleftarrow$ on timelines x, x' from (a) given by

$$x' \twoheadleftarrow x \quad \text{iff} \quad q \xrightarrow{b} q' \qquad \text{where } x(0) = (q, \alpha) \text{ and } x'(0) = (q', \alpha').$$

If we agree that

$$\twoheadleftarrow, x \models_{\mathfrak{l}} \mathsf{may}(\varphi) \quad \text{iff} \quad (\exists x' \twoheadleftarrow x) \ \twoheadleftarrow, x' \models_{\mathfrak{l}} \varphi$$

then $b(\Box, \boxed{p})$ portrays $\mathsf{may}(p)$,[8] assuming the obvious extension of the notion of portrayal in the previous section to b-strings. (For a smooth generalization of the total timeline-valuation pairs mentioned in section 2, we must restrict the range of $\mathfrak{l}$ to subsets α of P such that $(\forall p \in P)$ $p \in \alpha$ iff $\overline{p} \notin \alpha$.)

[8]How does may apply to the counterfactual in (13) above? The modal branches forward in time, leaving the work of moving back (into the past) to the perfect (expressed by *have -en*), as in Condoravdi (2002). We have, for simplicity, confined ourselves in this paper to future operators. An obvious way to introduce past operators would double the domain of a timeline from $\mathbb{N}$ to $\mathbb{N} \cup \{-i \mid i \in \mathbb{N}\}$.

What strings can we recover from a b-string? We can strip off the branches in a b-string $\mathsf{s} \in \Sigma^b$ to form the string $\mathsf{s}_{-b} \in \Sigma^+$, setting $\alpha_{-b} = \alpha$, $(\mathsf{ss}')_{-b} = (\mathsf{s}_{-b})(\mathsf{s}'_{-b})$, and $b(\mathsf{s},\mathsf{s}')_{-b} = \mathsf{s}_{-b}$. We may also collect all possibilities in the language $\pi(\mathsf{s}) \subseteq \Sigma^+$ including s_{-b} alongside the branching possibilities $\hat{\pi}(\mathsf{s})$ of s

$$\begin{array}{ll} \pi(\alpha) = \{\alpha\} & \hat{\pi}(\alpha) = \emptyset \\ \pi(\mathsf{ss}') = (\mathsf{s}_{-b})\pi(\mathsf{s}') + \hat{\pi}(\mathsf{s}) & \hat{\pi}(\mathsf{ss}') = (\mathsf{s}_{-b})\hat{\pi}(\mathsf{s}') + \hat{\pi}(\mathsf{s}) \\ \pi(b(\mathsf{s},\mathsf{s}')) = (\mathsf{s}_{-b})\pi(\mathsf{s}') + \pi(\mathsf{s}) & \hat{\pi}(b(\mathsf{s},\mathsf{s}')) = (\mathsf{s}_{-b})\pi(\mathsf{s}') + \hat{\pi}(\mathsf{s}). \end{array}$$

Let us call a set of b-strings a *b-language*, and agree that a b-language is *regular* if it is the set of b-strings accepted by a finite b-automaton.

Proposition 4. *If L is a regular b-language, then the languages $L_{-b} = \{\mathsf{s}_{-b} \mid \mathsf{s} \in L\}$ and $L_\pi = \bigcup\{\pi(\mathsf{s}) \mid \mathsf{s} \in L\}$ are both regular.*

Proof. Given a finite b-automaton for L, we need only drop the relation $\xrightarrow{b}$ to get a finite automaton for L_{-b}, and turn $s \xrightarrow{b} s'$ to $s \xrightarrow{\epsilon} s'$ for L_π. ⊣

The elimination of realis distance $\xrightarrow{b}$ in Proposition 4 takes us back to the "disconnection from the present" in CTL*.

References

Asher, N. and A. Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Beaver, D. and C. Condoravdi. 2003. A uniform analysis of *before* and *after*. In *Proc. Semantics and Linguistic Theory XIII*. Cornell Linguistics Circle Publications.

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

Bennett, B. and A. Galton. 2004. A unifying semantics for time and events. *Artificial Intelligence* 153:13–48.

Blackburn, P., C. Gardent, and M. de Rijke. 1996. On rich ontologies for tense and aspect. In J. Seligman and D. Westerståhl, eds., *Logic, Language and Computation*, vol. 1, pages 77–92. CSLI Lecture Notes Number 58, Stanford.

Clarke, E.M., O. Grumberg, and D.A. Peled. 1999. *Model Checking*. MIT Press, Cambridge, MA.

Comon, H., M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. 2002. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata.

Condoravdi, Cleo. 2002. Temporal interpretation of modals: Modals for the present and for the past. In D. Beaver, S. Kaufmann, B. Clark, and L. Casillas, eds., *The Construction of Meaning*, pages 59–88. CSLI, Stanford.

Dowty, David R. 1979. *Word Meaning and Montague Grammar*. Reidel, Dordrecht.

Emerson, E. Allen. 1992. Temporal and modal logic. In J. v. Leeuwen, ed., *Handbook of Theoretical Computer Science*, vol. B: Formal Methods and Semantics, pages 995–1072. MIT Press.

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

Fernando, Tim. 2004b. Inertia in temporal modification. In *Proc. Semantics and Linguistic Theory XIV*, pages 56–73. Cornell Linguistics Circle Publications.

Fernando, T. and R. Nairn. 2005. Entailments in finite-state temporality. In *Proc. 6th International Workshop on Computational Semantics*, pages 128–138. Tilburg University.

Heinämäki, O. 1972. Before. In *Papers from the 8th Regional Meeting of the Chicago Linguistic Society*, pages 139–151. University of Chicago.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.

Nelken, Rani and Nissim Francez. 1996. Automatic translation of natural language system specifications. In *CAV '96: Proceedings of the 8th International Conference on Computer Aided Verification*, pages 360–371. Springer-Verlag.

Parsons, Terence. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press, Cambridge, MA.

Schubert, Lenhart. 2000. The situations we talk about. In J. Minker, ed., *Logic-Based Artificial Intelligence*, pages 407–439. Kluwer Academic Publishers, Dordrecht.

Thomason, Richmond. 1984. Combinations of tense and modality. In D. Gabbay and F. Guenthner, eds., *Handbook of Philosophical Logic*, pages 135–165. Reidel.

Veltman, Frank. 1996. Defaults in update semantics. *J. Philosophical Logic* 25:221–261.

Zimmermann, Thomas Ede. 2000. Free choice disjunction and epistemic possibility. *Natural Language Semantics* 8:255–290.