Finite State Temporality and Context-Free Languages

Derek Kelleher
Trinity College Dublin
kellehdt@tcd.ie

Carl Vogel
Trinity College Dublin
vogel@tcd.ie

Abstract

In the finite-state temporality approach, events in natural language semantics have been characterized in regular languages, with strings representing sequences of temporal observations. We extend this approach to natural language constructions which are not regular. Context-free constructions are detailed and discussed. Superposition, the key operator in the finite-state temporality approach is investigated for context-free languages. The set of context-free languages is found to not be closed under superposition. However, as with intersection, the superposition of a context-free language and a regular language results in a context-free language. Previous work on subsumption and entailment is inapplicable to context-free languages, due to the undecidability of the subset relation for context-free languages.

1 Introduction

In recent years, events have been encoded as strings of a regular language, where a symbol in the language represents a set of predicate logic formulae that hold at a particular temporal instant, and the order of the symbols is associated with temporal order. Temporal granularity is revealed by the "superposition" of languages, essentially forming a more specific representation by collecting together formulae that hold at the same time. The representation is supplemented by notions of subsumption and entailment, allowing comparisons of information content, logical soundness, and completeness.

However, some natural language constructions concerning events are difficult to represent in this regular framework. These constructions suggest a relationship between the frequency of events, similar to the dependency of symbol frequencies on other symbol frequencies found in many context-free languages ($L = a^n b^n$). Together with their increase in complexity over regular langauges, this makes context-free languages a natural area of interest in this field.

As an example, take the expression "A as often as B", where A and B can be thought of as events. This construction implies a frequency relationship between the occurrence of two events, where A occurs at least as many times as B, possibly more. These events do not have to occur in an ordered sequence, where a sequence of As are followed by a sequence of Bs, As and Bs can occur in any order as long as the overall frequncy relationship is maintained. To be accurate, we must also allow for "instants" of time that separate any occurrence of the events we are interested in. Representing event A as the symbol a, event B as the symbol b, and an instant of time in which events may be occurring, but which are not relevant to our analysis by $\square$, we get strings of the form $a\square^*b$, $a\square^*b\square^*b\square^*a$, $b\square^*b\square^*a\square^*a\square^*a$ etc. These strings form a context-free language.

Moens and Steedman (1988) highlight the complicated nature of the phrase "when". They suggest that "When A, B" implies not a strictly temporal relationship, but a causal one, making it a prime candidate for representation by a context-free language. Note that "When I swear, I put money into the swear jar" implies that if I swear twice, I put money into the jar twice, but not necessarily in a particular temporal order. I may swear twice during the day, and have to wait until I get home to put money in the jar.

More formal (and seemingly less natural) constructions such as "an equal number of times as", while rare, do have a place in more formal literature such as legal documents. This particular construction appears in locations as varied as “The Federal Code of the United States of America” and the bye-laws of the town of New Canaan,CT: “he shall choose alternates in rotation so that the alternates chosen by the Chairman shall be seated as nearly an equal number of times as is possible”.[1]

Our analysis is restriced to the case of there being a frequency relationship between two types of events. It should be noted that the addition of a third event would lead to strings characterised by languages that are not context-free, similar to the difference between $a^n b^n$ and $a^n b^n c^n$. While these constructions tend to seem less natural (“I cried as often as I laughed, and I laughed as often as I sang”), they cannot be discounted.

The above linguistic data, while by no means exhaustive, provides a steady base from which to explore context-free languages in a finite-state temporality framework. The ubiquity of the phrases “when” or “whenever” highlights the need for this extension, while their causal nature, as opposed to temporal nature, suggests further ontological applications.

2 Background

Envisaging events as a sequence of “snapshots”, Fernando (2004) has encoded event-types as regular languages, made up of symbols representing sets of “fluents”(Φ), similar to those found in McCarthy and Hayes (1969). As well as representing event types, a regular language can represent sequences of temporal observations. The diagram below represents these two concepts:

$$L = \boxed{\sim swim(john, x)} \quad \boxed{swim(john, x)}^+ \quad \boxed{swim(john, m)}$$

$$L' = \qquad\qquad \boxed{mile(m)}^*$$

The “superposition” of two langauges is the componentwise union of their strings:

$$L\&L' = \bigcup_{k \geq 1} \{(\alpha_1 \cup \alpha'_1) \ldots (\alpha_k \cup \alpha'_k) \mid \alpha_1 \ldots \alpha_k \in L \text{ and } \alpha'_1 \ldots \alpha'_k \in L'\}$$

Intuitively, snapshots taken at the same temporal instant are merged, forming a larger picture of the world at that time:

$$\boxed{\sim swim(john, x) mile(m)}\boxed{swim(john, x) mile(m)}^+\boxed{swim(john, m) mile(m)}$$

The set of regular languages is closed under superposition ensuring that the superposition operation does not take us to a higher level of complexity(Fernando (2003)). Superposition allows us to define a reflexive, transitive relation (a pre-order) associated with the concept of subsumption. To preserve reflexivity subsumption $\unrhd$ is defined by:

$$L \unrhd L' \quad \text{iff} \quad L \subseteq L'$$

Subsuption can be thought of as relating to “information content”. A language that subsumes another is more specific than that language. It contains all the information of the other language, and more.

3 Superposition and Context-Free Languages

Superposition, as the central operation in the finite-state temporality framework, must be re-examined in light of our inclusion of context-free languages. The key question is whether the result of superposing a context-free language with either a regular language or another context-free language, is itself regular, context-free, or otherwise.

[1] http://ecode360.com/9045062 - accessed on 30/11/12.

Proposition 1 *The set of context-free languages is not closed under superposition.*

Proof(by counter-example): Let the set $\{\phi\}$ be represented by the symbol $\boxed{\phi}$, and the set $\{\psi\}$ be represented by the symbol $\boxed{\psi}$. The language $L_1 = \boxed{\phi}^n\boxed{\psi}^n$ is context-free, as is the language $L_2 = \boxed{\phi}^m\boxed{\psi}^{2m}$. L_1 is given by the grammar:

$S \quad \rightarrow \boxed{\phi}S\boxed{\psi}$
$S \quad \rightarrow e$

and L_2 by the grammar:

$S \quad \rightarrow \boxed{\phi}S\boxed{\psi}\boxed{\psi}$
$S \quad \rightarrow e$

The superposition of these two languages will contain strings consisting of three possible symbols: $\{\phi\} \cup \{\phi\} = \{\phi\}$ represented as $\boxed{\phi}$, $\{\phi\} \cup \{\psi\} = \{\phi, \psi\}$ represented as $\boxed{\phi\psi}$, and $\{\psi\} \cup \{\psi\} = \{\psi\}$ represented as $\boxed{\psi}$.

Strings in the language L_1 have length 2n, and strings in the language L_2 have length 3m. Strings can only be superposed if they have equal length, therefore only strings of length 6r from both languages can be superposed, resulting in strings of the same length. Strings in L_1 will consist of 3r $\boxed{\phi}$s followed by 3r $\boxed{\psi}$s, and strings in L_2 will consist of 2r $\boxed{\phi}$s followed by 4r $\boxed{\psi}$s. The superposition of these two strings will consist of 2r $\boxed{\phi}$s superposed with $\boxed{\phi}$s, r $\boxed{\phi}$s superposed with $\boxed{\psi}$s, and 3r $\boxed{\psi}$s superposed with $\boxed{\psi}$s, resulting in strings of the form $\boxed{\phi}^{2r}\boxed{\phi\psi}^{r}\boxed{\psi}^{3r}$. Introducing a homomorphism from $\boxed{\phi}$ to 'a', from $\boxed{\phi\psi}$ to 'b', and from $\boxed{\psi}$ to 'c', we have an equivalent language $a^{2r}b^rc^{3r}$.

If this language were context-free, given that it is infinte, there would be some constant K such that any string longer than K would be representable as a string uvxyz such that v and y are not empty and are pumpable. If we choose the string $a^{2K}b^Kc^{3K}$ as a string longer than K, we should be able to factorize it in this manner. If we chose v to have both as and bs or both bs and cs, then upon pumping it, we would have bs before as or cs before bs, which would result in a string not in our language. The same considerations apply to choosing y. Therefore v and y must each contain only as, or only bs, or only cs. Pumping v and y would therefore increase the number of one or two of the symbols but not all three, thereby losing the frequency relationship between the three symbols. The language cannot be context-free.□

Proposition 2 *The superposition of a context-free language with a regular language is context-free.*

Proof: Given L_1, a context-free language, and L_2, a regular language, let $P = \langle Q_P, \Sigma, \Gamma, \Delta_P, q_{P0}, F_P\rangle$ be a pushdown-automaton accepting L_1 and $A = \langle Q_A, \Sigma, \delta_A, q_{A0}, F_A\rangle$ be a finite-state-automoton accepting L_2. Δ_P is the set of transitions of the form $(q_i, a, A) \rightarrow (q_j, \gamma)$ interpreted as: when in state q_i , with input symbol a, and symbol A at the top of the stack, go to state q_j and replace A by the string γ, and δ_A is the set of transitions of the form $(q_i, a) \rightarrow (q_j)$ interpreted as: when in state q_i with input symbol a, go to state q_j. We form a pushdown automaton $R = \langle Q_P \times Q_A, \Sigma, \Gamma, \Delta_{P\times A}, (q_{P0}, q_{A0}), F_P \times F_A\rangle$, with transitions $\Delta_{P\times A}$ constructed as follows:

1. If Δ_P contains a rule of the form $(q_0, e, e) \rightarrow (q_1, S)$, then $\Delta_{P\times A}$ contains a rule of the form $((q_0, q_0), e, e) \rightarrow ((q_1, q_0), S)$.

2. If Δ_P contains a rule of the form $(q_1, e, A) \rightarrow (q_1, \gamma)$, then $\Delta_{P\times A}$ contains rules of the form $((q_1, q_x), e, A) \rightarrow ((q_1, q_x), \gamma)$ for every $q_x \in Q_A$.

3. If Δ_P contains a rule of the form $(q_1, a, a) \rightarrow (q_1, e)$, then $\Delta_{P\times A}$ contains rules of the form $((q_1, q_x), a \cup b, a) \rightarrow ((q_1, q_y), e)$ if and only if there is a transition $(q_x, b)\rightarrow(q_y)$ in δ_A.

The new transitions are akin to running the PDA and FSA in tandem, where a state (q_x, q_y), while strictly a state of R, can be thought to represent the simultaneous states of P and A. A rule of type 1 and rules of type 2 perform the same stack operations as the PDA they were derived from. Therefore, R can produce on its stack the same set of strings that P produces on its stack. No input symbol is being read while these stack operations are performed, therefore R should remain in state (q_x, q_y). Rules of type 3 ensure that if R is in a state (q_1, q_y) with an input symbol $a \cup b$, and encounters the terminal symbol a on its stack, along with there being a transition in A from q_y to q_z on input b, then R will transition to state (q_1, q_z), and delete a from its stack. These are exactly the states that P and A would seperately be in upon reading input a and b respectively. Thus, if P reads a string $a_1 \ldots a_n$ and is in a final state with an empty stack (i.e. P accepts this string), and A reads a string $b_1 \ldots b_n$ and is in a final state (i.e. A accepts this string), then R will be in a final state upon reading the superposition of these two strings. If P accepts a language L_1 and A accepts a language L_2, then R will accept $L = L_1 \& L_2$.□

If we superpose the context-free language that represents "I laughed as often as I cried" with the regular langauge that represents "an hour" to get a language representing "In an hour, I cried as often as I laughed", this language, as the superposition of a context-free langauge and a regular language, will be context-free.

4 Final Remarks

Further work will involve investigating how the concepts of subsumption and entailment can be related to context-free languages. In this framework, entailment is defined in terms of subsumption, which is defined in terms of the subset relation (Fernando and Nairn (2005)). However, according to Hopcroft et al. (1979), the problem of whether a context-free language is a subset of another context-free language is undecidable. If the subset relation cannot be calculated for context-free languages, subsumption and entailment relations break down.

One possible avenue of exploration is the making of regular approximations of context-free languages (Mohri et al. (2001)). This would preserve the subsumption and entailment relations, but at a possible cost to accurately representing the context-free construction, possibly losing the exact relationship between the frequencey of two symbols.

References

Fernando, T. (2003). Finite-state descriptions for temporal semantics. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5).*

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

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

Hopcroft, J., R. Motwani, and J. Ullman (1979). *Introduction to automata theory, languages, and computation*, Volume 2. Addison-wesley Reading, MA.

McCarthy, J. and P. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In M. Meltzer and D. Michie (Eds.), *Machine Intelligence 4*. Edinburgh University Press.

Moens, M. and M. Steedman (1988). Temporal ontology and temporal reference. *Computational Linguistics 14*(2), 15–28.

Mohri, M., M. Nederhof, et al. (2001). Regular approximation of context-free grammars through transformation. *Robustness in language and speech technology 17*, 153–163.