

# Quantification and Implication in Semantic Calendar Expressions Represented with Finite-State Transducers

Jyrki Niemi and Kimmo Koskenniemi

Department of General Linguistics, University of Helsinki

PO Box 9, FI-00014 University of Helsinki, Finland

{jyrki.niemi, kimmo.koskenniemi}@helsinki.fi

## Abstract

This paper elaborates a model for representing semantic calendar expressions (SCEs), which correspond to the intensional meanings of natural-language calendar phrases. The model uses finite-state transducers (FSTs) to mark denoted periods of time on a set of timelines represented as a finite-state automaton (FSA). We present a treatment of SCEs corresponding to quantified phrases (*any Monday*; *every May*) and an implication operation for requiring the denotation of one SCE to contain completely that of another.

## 1 Introduction

This paper elaborates the temporal representation model proposed in Niemi and Koskenniemi (2007) and developed in Niemi and Koskenniemi (2008). This *bracketing FST model* covers temporal information ranging from simple dates to such meanings as *6–8 pm on every Monday in April, except on Easter Monday*. The model represents *semantic calendar expressions* (SCEs) using finite-state transducers (FSTs) that bracket periods of time on timelines represented as a finite-state automaton (FSA). Motivations for a finite-state representation include an efficient treatment of periodicity and certain kinds of sparse sets of sets common in calendar information, as well as a well-known theory.

In this paper, we treat SCEs corresponding to quantified calendar phrases, such as *any Monday* and *every May*. We also present implication for representing such cases as a course with compulsory attendance, whose all class times should coincide with the free slots of time of a student.

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported license* (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

## 2 Semantic Calendar Expressions (SCEs) and the Bracketing FST Model

We use the semi-formal SCEs as the basic representation. An SCE typically corresponds to the disambiguated intensional meaning of a natural-language calendar phrase. An SCE may denote either a specific period of time or a set of periods. An SCE is compiled to a sequence of *bracketing FSTs* through regular (relation) expressions.

Table 1 lists some implemented SCE constructs and the corresponding calendar phrases. Most constructs can be combined with each other.

Construct	SCE example; calendar phrase
calendar period	may; fri; calday <i>May; Friday; (calendar) day</i>
list (union)	union (mon, fri, sun) <i>Monday, Friday and Sunday</i>
common part (intersection)	intersect (aug, y2008) <i>August 2008</i>
interval	interval (may, jun) <i>May to June</i>
anchored	nth_following (3, mon, easter) <i>the third Monday after Easter</i>
quantified	any_n (1, mon) <i>any (single) Monday</i>

Table 1: Examples of SCE constructs and the corresponding calendar phrases

The bracketing FST model represents the denotations of an SCE as an acyclic *timeline FSA* defining a finite *timeline string* for each alternative denotation.<sup>1</sup> A *basic timeline FSA* defines a single timeline consisting of brackets and labels for calendar periods, with no denotations marked. The following is a (simplified) timeline for the year 2008 at month level (spaces separate symbols):

<sup>1</sup>Timeline FSAs are equivalent to the timeline FSTs of Niemi and Koskenniemi (2008).

[y y2008 [m Jan ]m [m Feb ]m ... [m Dec ]m ]y

To represent the denotation of an SCE  $s$ , a basic timeline FSA is composed with a bracketing FST (sequence) representing  $s$ . The result is a timeline FSA with the denotation marked with *marker brackets* {in ... }in. The following timeline corresponds to union (jan, mar) (*January and March*):

[y y2008 {i3 {i1 [m Jan ]m }i1 }i3 [m Feb ]m {i3 {i2 [m Mar ]m }i2 }i3 [m Apr ]m ... [m Dec ]m ]y

First, January is marked with  $i1$  and March with  $i2$ . Then the bracketing FST for union marks each period  $i1$  and  $i2$  with  $i3$ , the denotation of the whole.

### 3 Representing Quantified SCEs

A natural-language calendar phrase may be unquantified, such as *Monday*, or it may contain different kinds of quantification, such as *all Mondays*, *any Monday*, *three Mondays* and *all Mondays in some May*.<sup>2</sup> We treat *any Monday*, *some Monday* and *one Monday* as meaning any single Monday, and we equate *all Mondays*, *each Monday* and *every Monday*. Numerical quantification can be generalized to intervals, possibly open-ended, such as *two to five Mondays* and *at most four Mondays*.

#### 3.1 Collective Representation (All)

The meaning *all Mondays* is represented simply as the SCE  $\text{mon}$ , which is represented as a timeline FSA with each Monday marked on the same timeline, as illustrated in Fig. 1 (simplified).<sup>3</sup> We call this the *collective representation* of an SCE.

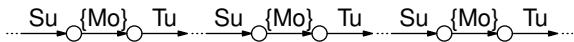


Figure 1: A timeline FSA for the SCE  $\text{mon}$  (*all Mondays*); “{Mo}” denotes a marked Monday

The denotation can be interpreted in several ways: as the union of all Mondays, as all the parts of a timeline that are part of a Monday, as the recurring “event” of Monday, or as a disconnected (non-convex) interval of all Mondays.

#### 3.2 Distributive Representation (Any)

The meaning *any Monday* is represented as the SCE  $\text{any}_n(1, \text{mon})$  and as a timeline FSA with

<sup>2</sup>All could be regarded as universal quantification, and *any* resembles existential quantification, but because of some differences, we avoid using these logical terms.

<sup>3</sup>Each transition in the figures corresponds to a number of states and transitions between them in the actual timeline FSA, as the representation of each day consists of the calendar day brackets, symbols for the day of the week and the day of the month, and possibly finer granularities inside.

each Monday marked on a separate, alternative timeline of its own, as illustrated in Fig. 2. We call this the *distributive representation* of an SCE.<sup>4</sup>

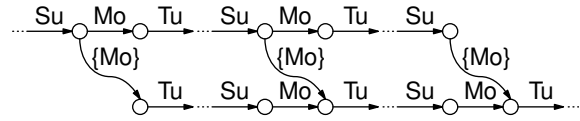


Figure 2: A timeline FSA representing the SCE  $\text{any}_n(1, \text{mon})$  denoting any single Monday

For numerical quantification, we combine collective and distributive representation. For example,  $\text{any}_n(3, \text{mon})$  (*three Mondays*) is represented as a timeline FSA defining a set of timelines, each with three Mondays marked, so that the set covers all possible combinations. A collective representation alone would not suffice: a single timeline with all periods of three Mondays marked would in effect represent all Mondays.

#### 3.3 Collective Representation as Primary

In natural language, an unquantified calendar phrase, such as *Monday*, is typically underspecified and refers to the closest preceding or following Monday relevant in the context. In the bracketing FST model, however, we represent unquantified expressions collectively: the SCE  $\text{mon}$  represents both *Monday* and *all Mondays*.

A major practical reason for preferring the collective representation is that it is easy to construct a bracketing FST that splits a single timeline with each Monday marked to a set of alternative timelines, each with only one of the Mondays marked, whereas the converse operation is not possible. In effect, a bracketing FST can only refer to symbols on a single timeline at a time.

A single timeline with all Mondays marked can also be used as a basis for such an SCE as  $\text{nth\_following}(3, \text{mon}, \text{easter})$  (*the third Monday after Easter*). The bracketing FST corresponding to  $\text{nth\_following}$  gets as its input a timeline with each Monday and each Easter marked. It then counts the third Monday after each Easter and marks it. This would be much more difficult with each Monday marked on a timeline of its own.

#### 3.4 Combining Quantified SCEs

Multiple SCEs with quantification can be combined appropriately. For example, *all Mondays in*

<sup>4</sup>We deviate from the common terminology that uses *collective* for *all Mondays* and *distributive* for *each Monday*.

any (single) May is represented by a timeline FSA defining a set of timelines, each with all Mondays of a single May marked, and any (single) Monday in every May by one with a single Monday marked in each May, covering all possible combinations.

Combinations of quantified SCEs can often be represented compositionally with intersection. For example, *all Mondays in every May* is represented as `intersect (mon, may)`, *all Mondays in any May* as `intersect (mon, any_n (1, may))` and *any Monday in any May* as `intersect (any_n (1, mon), any_n (1, may))`.

However, *any Monday in every May* cannot be represented simply with intersection, since intersecting a set of timelines, each with only one Monday marked, and a timeline with every May marked would result in timelines with one Monday marked in (at most) one May. For this case, we have defined the SCE operation `n_within_each (n, period, frame)`, which marks *n periods* within each *frame*. *Any Monday in every May* is then represented as `n_within_each (1, mon, may)`.

### 3.5 Other Uses of Distributive Representation

A distributive representation is obligatory for an SCE denoting possibly overlapping periods of time, although each alternative timeline may contain several non-overlapping periods.

For example, we represent `n_consecutive (3, calday)` (*three consecutive days*) as a set of timelines, each with one possible combination of non-overlapping periods of three consecutive days marked.<sup>5</sup> If each timeline had only a single period marked, it would complicate representing such an SCE as union (`intersect (n_consecutive (3, calday), may)`, `intersect (n_consecutive (3, calday), jun)`) (*three consecutive days in May and in June*) so that the periods of consecutive days in May and June are marked on the same timeline. In contrast, a single timeline with all possible non-overlapping periods marked would not cover the periods overlapping with the marked ones.

A distributive representation is also used for SCEs containing a distributive union operation to represent a disjunctive meaning. For example, `distr_union (union (mon, fri), union (tue, sun))`

<sup>5</sup>Since consecutive days are adjacent, they can be enclosed in marker brackets and treated as a single connected period, with several periods on a single timeline. In contrast, the multiple disconnected periods of *three consecutive Mondays* cannot be represented in a general way on the same timeline in the bracketing FST model, but only by having a different marker bracket index for each period of three Mondays.

(*Monday and Friday or Tuesday and Sunday*) is represented as a timeline FSA defining two timelines, one with every Monday and Friday marked and the other with every Tuesday and Sunday.

### 3.6 Distributive Representation and First

Distributive representation requires special considerations in conjunction with some SCE operations, most notably `nth_within (n, period, frame)`, which marks the *nth period* within each longer period *frame*. Although a *period* might be the *nth* marked one within a *frame* on one timeline, alternative timelines might contain earlier occurrences.

As an example, we consider `nth_within (1, n_consecutive (3, workday), jun)` (*the first period of three consecutive working days in June*).<sup>6</sup> For a June beginning on a Sunday, we have alternative timelines with the first period of three working days beginning on Monday, Tuesday and Wednesday, but we would like to mark only the one beginning on Monday. However, a bracketing FST cannot refer to the alternative timelines to test if any of them contains an earlier applicable period.

As a solution, we have such operations as `n_consecutive insert an alternative marker bracket` (denoted by `[. . .]` below) on each timeline into each position in which it adds a marker bracket (`{. . .}`) on another timeline. The following simplified alternative timelines illustrate the example above:

Su { Mo [ Tu [ We } Th ] Fr ] Sa

Su [ Mo { Tu [ We ] Th } Fr ] Sa

Su [ Mo [ Tu { We } Th ] Fr ] Sa

The operation `nth_within` seeks the first marked period in June with no opening alternative marker bracket between it and the beginning of June.

## 4 Implication: All or Nothing

In some applications, an SCE may denote a set of periods of times all of which should be contained in those denoted by another SCE, or if impossible, none of them should be. For example, all the class times of a course with compulsory attendance should coincide with the free slots of time in the calendar of a student wishing to attend the course. An intersection of the class times and the student's free slots of time would also contain partial results if he or she could attend only some of the classes.

<sup>6</sup>The SCE `nth_within (1, n_consecutive (3, calday), jun)` (*the first period of three consecutive (calendar) days in June*) denotes the same as `first_n_within (3, calday, jun)` (*the first three days in June*). A similar rephrasing would not be correct for working days, however.

To obtain the desired result, we use the operation  $\text{impl}(a, b)$  to mark all the periods  $a$  if and only if they all are fully contained in the periods  $b$ . If a point of time is in  $a$ , it must also be in  $b$  for  $a$  to be marked, so the operation can be regarded of as a kind of an implication  $a \rightarrow b$ . Above, we would compute  $\text{impl}(\text{course}, \text{student\_free})$ .

A course with alternative instances would be represented with a timeline FSA defining an alternative timeline for each instance. The bracketing FST corresponding to the above implication would then mark the class times of each instance that is completely within the student’s free slots of time.

## 5 Related Work

We briefly mention the approaches to quantification of some research related to the bracketing FST model in purpose or coverage. TEL (Endriss, 1998) represents universally quantified expressions like unquantified ones. TEL has numerical quantification and quantifier negation. TCNL (Han and Lavie, 2004) represents universal quantification as an enumeration. An unquantified expression denotes an underspecified time. Like TEL, TCNL has no explicit existential quantification. Ohlbach and Gabbay (1998) represent universal quantification with a parametrized modal operator “always within a period” and existential with “sometime within a period”. Cukierman and Delgrande (1998) represent quantified expressions in a way resembling that of ours but unquantified expressions effectively as existentially quantified. TimeML (Saurí et al., 2006) represents quantification by quantifier and frequency attributes. OWL-Time (Pan and Hobbs, 2005) uses temporal aggregates for universal and numerical quantification.

## 6 Discussion and Further Work

In our view, SCEs corresponding to typical quantified calendar phrases can be represented in the bracketing FST model fairly naturally, although the naturalness of representing unquantified, underspecified phrases collectively can be disputed, and the representation of *any Monday in every May* is not compositional. Implication, in turn, would seem useful for representing a set of periods of time fully contained in another set of periods.

Although some types of calendar information are impossible to represent exactly or naturally with finite-state methods, we find the bracketing FST model a promising representation for many

common types. However, to be usable in practice, the model needs further work in both coverage and efficiency. Moreover, applications would benefit from a component to parse a (restricted) natural-language calendar phrase to an SCE and another one to generate the former from the latter.

## Acknowledgements

This paper represents independent work by the first author based on the suggestions of the second author and funded by the Graduate School of Language Technology in Finland. We thank the anonymous reviewers for their valuable comments.

## References

- Cukierman, Diana and James P. Delgrande. 1998. Expressing time intervals and repetition within a formalization of calendars. *Computational Intelligence*, 14(4):563–597.
- Endriss, Ulrich. 1998. Semantik zeitlicher Ausdrücke in Terminvereinbarungsdialogen. *Verbmobil Report 227*, Technische Universität Berlin, Fachbereich Informatik, Berlin, August.
- Han, Benjamin and Alon Lavie. 2004. A framework for resolution of time in natural language. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1):11–32, March.
- Niemi, Jyrki and Kimmo Koskenniemi. 2007. Representing calendar expressions with finite-state transducers that bracket periods of time on a hierarchical timeline. In Nivre, Joakim, Heiki-Jaan Kaalep, Kadri Muischnek, and Mare Koit, editors, *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007*, pages 355–362, Tartu, Estonia. University of Tartu.
- Niemi, Jyrki and Kimmo Koskenniemi. 2008. Representing and combining calendar information by using finite-state transducers. In *Proceedings of the Seventh International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP) 2008*. To appear.
- Ohlbach, Hans Jürgen and Dov Gabbay. 1998. Calendar logic. *Journal of Applied Non-classical Logics*, 8(4):291–324.
- Pan, Feng and Jerry R. Hobbs. 2005. Temporal aggregates in OWL-Time. In *Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2005)*, pages 560–565, Clearwater Beach, Florida. AAAI Press.
- Saurí, Roser, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. TimeML annotation guidelines, version 1.2.1. [http://timeml.org/site/publications/timeMLdocs/annguide\\_1.2.1.pdf](http://timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf), January.