Towards modeling the semantics of calendar expressions as extended regular expressions

Jyrki Niemi and **Lauri Carlson**
Department of General Linguistics
University of Helsinki
P.O. Box 9, FI-00014 University of Helsinki, Finland
{jyrki.niemi,lauri.carlson}@helsinki.fi

Abstract

This paper proposes modeling the semantics of natural-language calendar expressions as extended regular expressions (XREs). The approach covers expressions ranging from plain dates to such ones as *the second Tuesday following Easter*. The paper presents basic calendar XRE constructs, sample calendar expressions with their representations as XREs, and possible applications in reasoning and natural-language generation.

1 Introduction

Temporal information and calendar expressions are essential in various applications, for example, in event calendars, appointment scheduling and timetables. The information should often be both processed by software and presented in a human-readable form.

Calendar expressions denote periods of time that do not depend on the time of use of the expression. They can be plain dates, times of the day, days of the week or combinations of them, or they can be more complex, such as *the second Tuesday following Easter*. The denotation may be underspecified or ambiguous without context.

In this paper, we propose modeling the semantics of natural-language calendar expressions as extended regular expressions (XREs). The well-known semantics of XREs can be used in reasoning with the temporal information so represented. We also believe that XREs are structurally so close to the corresponding natural-language expressions that the latter can be generated from the former fairly straightforwardly.

The rest of the paper is organized as follows. Section 2 outlines a string-based model of time. Section 3 presents a number of calendar expression constructs and the corresponding XREs. Section 4 briefly describes experiments on reasoning with calendar XREs and natural-language generation from them. Section 5 mentions some related work in temporal expression research. Section 6 concludes with discussion and some directions for further research.

2 A string-based model of time

To be able to represent calendar expressions as XREs, we represent periods of time as strings of symbols. The model outlined here is only one possible one.

Time can be modelled as an infinite timeline. We choose a finite subset of the infinite timeline and partition it into a finite number of consecutive basic periods, for example minutes or seconds. For each basic period t_i, there is a unique corresponding symbol a_i in the alphabet Σ of calendar XREs, as illustrated in Figure 1. The string $a_1 \ldots a_n$ corresponds to the finite subset of the timeline; we denote it as T.

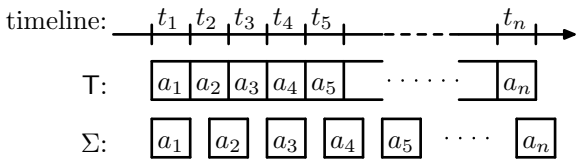

Figure 1: The relationship of the timeline, the timeline string T and the XRE alphabet Σ.

A single calendar XRE defines a regular language (a set of symbol strings) corresponding to a set of possibly disconnected periods of time. An XRE representing an inconsistent temporal expression, such as *30 February*, denotes the empty set.

S. Werner (ed.), *Proceedings of the 15th NODALIDA conference, Joensuu 2005*, Ling@JoY 1, 2006, pp. 133–138
ISBN 952-458-771-8, ISSN 1796-1114. http://ling.joensuu.fi/lingjoy/

The language of an XRE may contain strings that are not substrings of the timeline string T. They typically result from a concatenation or a complement operation. Such a string contains a symbol a_i followed by a_j, where $i \geq j$. It is not meaningful as a representation of time, as a period of time may not be followed by the same or a preceding period. We can limit the languages of XREs to meaningful periods by intersecting them with the set of (possibly disconnected) substrings of T. However, we leave the intersection implicit in the examples in this paper.

3 Calendar expressions and their XREs

3.1 General features

The calendar expression constructs presented in this section demonstrate key features of calendar XREs. The constructs were found in a corpus of Web pages. Table 1 lists examples of the calendar expression constructs that we have treated. A description of the constructs omitted from this paper and more details of the described ones can be found in Niemi (2004).

12.00; September; year; Easter
4 to 10 hours; 3 weeks short of 2 years
Mondays and Wednesdays
Mon and Wed or Tue and Thu
on Friday; in September
Christmas Eve falling on a Friday
22 May; in 2005 by April
a weekend containing Christmas Day
Monday to Friday; before 15 May
from 10 am on Sunday to 6 pm on Friday
8 am, except Mondays 9 am
every day except Monday
the second Tuesday following Easter
four weeks before Christmas
the weekend preceding a mid-week Christmas
on three consecutive Sundays
six consecutive months from May 2002
the third and fourth Wednesday of the month
every second Wednesday of the year
even Tuesdays of the month
four weeks a year
two Saturdays a month during the school year

Table 1: Examples of calendar expressions.

We are mainly interested in the semantics of calendar expressions, abstracted from different syntactic variants. We assume the semantics to be mostly language-independent.

We generally present the expressions in a simple form, without considering special cases that might complicate the required XRE.

We have tried to make the calendar XRE constructs compositional, so that they would combine with each other analogously to the corresponding natural language calendar expressions. However, a number of constructs are compositional only to a limited extent or not at all.

3.2 Regular expression operations

To construct more complex calendar XREs from basic ones, we use a number of regular expression operations. They include concatenation (.), union ($\cup$) and Kleene star (*); intersection ($\cap$), complement ($\neg$) and difference ($-$); and substring, quotient and affix operations. (Notations are explained where they are used.) The last three types of operations extract parts of their operand strings, so they are defined by means of regular relations (finite transducers).

Exponentiation is a notational shorthand denoting generalized concatenation power: A^N denotes the expression A concatenated N times, where N may be either a single natural number or a set of natural numbers.

We also use simple parametrized macros to simplify XREs containing repeating subexpressions and to make temporal XREs structurally closer to the corresponding natural-language expressions. The macros have no recursion or other means of control.

3.3 Basic calendar expressions

The basic expressions of calendar XREs denote sets of calendar periods, such as a day, month or year. An unqualified natural-language calendar expression, such as *Monday* or *January*, typically refers to the nearest past or future period relevant in the context. In this work, however, we interpret such expressions as underspecified, for example, referring to any Monday. The calendar XRE corresponding to *Monday* is Mon, which denotes the set of all Mondays.

The basic expressions correspond to the basic periods of the Gregorian calendar. They are represented as predefined constant sets of substrings of the timeline string. These include both generic periods, such as day, month and year (min to year), and specific ones, such as

each hour (h00 to h23), day of week (Mon to Sun), day of month (1st to 31st), month (Jan to Dec) and year (y$nnnn$). Hours and shorter units of time are also treated as periods; for example, hour 10 is the hour beginning at 10 am. We also assume appropriately predefined sets for seasons and holidays, such as Easter and Christmas_Day.

The generic calendar periods are unions of the corresponding specific calendar periods; for example, a week is any connected seven-day period from a Monday to the following Sunday. However, a week may also be a duration or a measurement unit of any seven days or 168 hours, possibly disconnected. A variable-length duration, such as a month, is represented as the union of the possible lengths.

Figure 2 illustrates the relationship of basic calendar periods to the timeline string T.

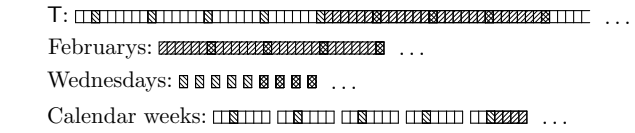

Figure 2: Basic calendar periods related to a timeline string T, assuming a year that begins with a Monday.

3.4 Basic combining constructs

Four basic constructs combining basic calendar expressions are lists, concatenation, refinement and intervals.

Lists of calendar expressions are in general represented using union. For example, *Mondays and Wednesdays* is represented as the XRE Mon $\cup$ Wed, meaning "any single Monday or Wednesday".

Concatenation juxtaposes periods of time. Concatenating non-adjacent periods results in a disconnected period; for example, *two Sundays* is represented as Sun . Sun.

If a calendar expression contains both *and*s and *or*s, we use concatenation for *and* and union for *or*: for example, *Mon and Wed or Tue and Thu* is represented as (Mon . Wed) $\cup$ (Tue . Thu).

Refinement combines periods of different lengths to more specified expressions using intersection and the substring operation $\mathbf{in}_+$. Figure 3 illustrates refinement with an XRE representing the expression *20 May*. First, any periods of time of any May are represented using the substring operation: $\mathbf{in}_+$ May. The resulting set is then intersected with the set 20th representing any 20th days of a month, yielding exactly those periods of any May that correspond to a 20th day of the month: 20th $\cap$ $\mathbf{in}_+$ May.

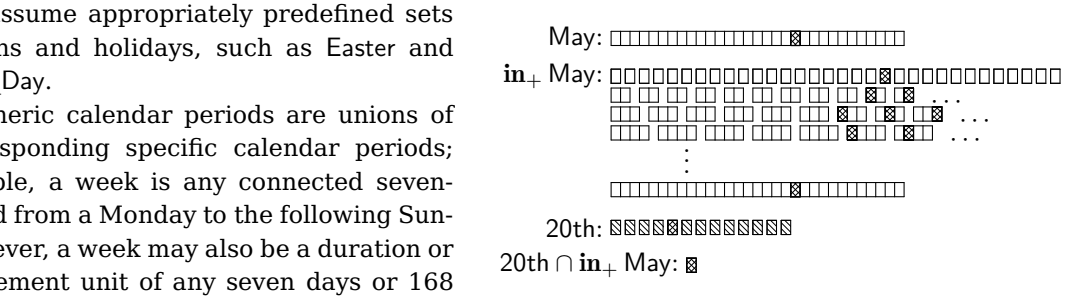

Figure 3: Constructing the XRE for the calendar expression *20 May*.

An interval *Monday to Friday* begins from a Monday and almost always ends to the closest following Friday. This interval can be expressed as the calendar XRE Mon . $\neg$ (Σ^*.Mon.Σ^*) . Fri, or, using the right closure shorthand operation $\triangleright$, as Mon$\triangleright$. Fri. Both XREs read as "a Monday followed by anything not containing a Monday, followed by a Friday".

3.5 More complex calendar expressions

In this subsection, we present exception expressions, anchored expressions and ordinal expressions as examples of more complex types of calendar expressions.

The expression *8 am, except Mondays 9 am* is an exception expression, where *8 am* is a default time, *Mondays* an exception scope and *9 am* an exception time (cf. Carlson (2003)). This can be expressed in XREs using union, difference and intersection: (h08 $-$ $\mathbf{in}_+$ Mon) $\cup$ (h09 $\cap$ $\mathbf{in}_+$ Mon). If the exception time is omitted, the difference alone suffices.

An anchored expression, such as *the second Tuesday following Easter*, denotes a time relative to an anchor time. The expression denotes the last day in a string of days beginning from Easter, containing exactly two Tuesdays and ending in a Tuesday. Using the closure operation, it can be expressed as the XRE (Easter.(Tue$\triangleright$)^2)\\Tue. The suffix operation $A\backslash\backslash B$ denotes the B at the end of A. We have defined for this construct a macro, with which the XRE would be nth_following(2,Tue,Easter). Similar *preceding*-expressions can be represented by changing the order of the elements and the direction of the closure and affix operations.

The expression *every second Wednesday of the year* is an ordinal expression. We interpret it as denoting the first, third, fifth and so on, Wednesday of the year. We represent it as the XRE $((((\triangleleft\mathsf{Wed})^2)^*.\triangleleft\mathsf{Wed}) \cap \mathbf{pref}_+\ \mathsf{year}) \backslash\backslash \mathsf{Wed}$. The left closure $\triangleleft A$ is a shorthand for $\neg(\Sigma^*.A.\Sigma^*).A$, and the untyped prefix operation $\mathbf{pref}_+\ A$ denotes all non-empty prefixes of A. Since this XRE construct contains a concatenation power inside a Kleene star, it counts multiples of a natural number larger than one, and thus it is not star-free (McNaughton and Papert, 1971). The only other non-star-free type of calendar expressions that we have encountered are parity expressions, such as *even Tuesdays of the month*. They can be represented similarly.

4 Application experiments

We have briefly experimented on temporal reasoning with calendar XREs, and on generating corresponding natural-language expressions from them.

4.1 Temporal reasoning with calendar XREs

We have mainly considered a form of temporal reasoning that finds common periods of time denoted by two calendar XREs, which can be used in querying temporal data. For example, a query to an event database could be used to find out which museums are open on Mondays in December, or at what time a certain museum is open on Mondays in December. For the former query, we should find the set of periods of time common to the query XRE and each target XRE, and for the latter, whether they have common periods or not. Both require basically computing the intersection of the query and target XREs.

In principle, such reasoning could be implemented straightforwardly as model checking, by constructing finite-state automata from the XREs and intersecting them, and by either enumerating their languages or checking if the intersection is empty. In practice, however, constructing the automata would often require too much space or time or both to be tractable. Moreover, the resulting language as such is usually not desirable as the result, as it may be very large and incomprehensible to a human.

We have used the Xerox Finite-State Tool (XFST) (Karttunen et al., 1997) to experiment with XREs and with reasoning based on model-checking.

4.2 Natural-language generation from calendar XREs

Calendar XREs could be used as a language-independent input formalism for calendar expressions in a possibly multilingual natural-language generation system. Our hypothesis was that calendar XREs should be structurally close enough to the corresponding natural-language expressions to make simple generation feasible. We thus experimented with a simple XSLT-based natural-language generation component for calendar XREs.

We encountered more complexities in our experiments than we had expected, but they were at the surface-syntactic and morphological level, not in higher-level structures. The use of XRE macros was essential; without them, the natural-language expressions generated from complex XREs would have been cumbersome and their intended meaning probably impossible to understand.

We simplified the generation component proper by assuming it to be preceded by a separate transformation phase. This phase could, for example, reorder or regroup the subexpressions of an XRE while preserving the meaning of the whole. For instance, it could transform *on Mondays in December* to *in December on Mondays*, or *1 May to 25 May* to *1–25 May*.

5 Related work

Temporal expressions in general have been much studied, including modeling and reasoning with the semantics of calendar expressions. Our main inspiration has been Carlson's (2003) event calculus, a part of which is modeling calendar expressions as XREs.

The Verbmobil project (Wahlster, 2000) had a formalism of its own to represent and reason with temporal expressions occurring in appointment negotiation dialogues (Endriss, 1998). Its coverage of natural-language calendar expressions was similar to that of calendar XREs, but it did not cover disconnected periods of time.

The calendar logic of Ohlbach and Gabbay (1998) can represent calendar expressions of various kinds. However, calendar logic expressions are not structurally as close to natural-language expressions as calendar XREs.

Han and Lavie (2004) use their own formalism in conjunction with reasoning using temporal constraint propagation. They explicitly cover more types of expressions than we do, including underspecified and quantified expressions, such as *every week in May*.

Regular expressions are used in conjunction with temporal expressions by Karttunen et al. (1996) who express the syntax of dates as regular expressions to check their validity. They limit themselves to rather simple dates, however.

Fernando (2002; 2004) uses regular expressions to represent events with optional temporal information. Focusing on events, his examples feature only simple temporal expressions, such as *(for) an hour*.

6 Discussion and further work

In our view, extended regular expressions would in general seem to be fairly well suited to modeling the semantics of calendar expressions. Calendar XREs are structurally relatively close to natural-language calendar expressions, and the semantics of regular expressions is well known. The former property can be of use in natural-language generation, the latter in reasoning. The approach can also be extended to cover a number of deictic and anaphoric temporal expressions.

We have applied XREs only to calendar expressions of the Gregorian calendar system, but we expect the representation to work with any calendar system based on similar principles of hierarchical periods, provided that appropriate basic calendar expressions have been defined.

Our main future research goal is to find a tractable and practical reasoning method, possibly processing XREs syntactically using term rewriting. A major drawback of term rewriting is that each different XRE operation should be separately taken into account in the rewriting rules. We could also try combining several different approaches, using each one where it is best.

While calendar XREs cover a large number of different types of calendar expressions, they cannot naturally represent certain kinds of expressions: fuzzy or inexact calendar expressions, such as *about 8 o'clock*; internally anaphoric expressions, such as *9.00 to 17.00, an hour later in winter*; or fractional expressions, such as *the second quarter of the year*. Extending the formalism to cover these expression types would be another major goal. The representation of fuzzy temporal expressions has been researched by for example Ohlbach (2004).

There are also a number of limitations in the compositionality of the calendar XRE constructs, and the XREs required for some types of calendar expressions are rather complex. In particular, an expression allowing disconnected periods of time can be significantly more complex than a similar one only working with connected periods. We would also like to try to treat these issues.

Lastly, we also intend to explore options to combine calendar XREs with event information, or at least to consider calendar expressions in their context. Such an approach might in some cases help resolve the meaning of a single fuzzy or underspecified calendar expression.

References

Lauri Carlson. 2003. Tense, mood, aspect, diathesis: Their logic and typology. Unpublished manuscript, February.

Ulrich Endriss. 1998. Semantik zeitlicher Ausdrücke in Terminvereinbarungsdialogen. Verbmobil Report 227, Technische Universität Berlin, Fachbereich Informatik, Berlin, August.

Tim Fernando. 2002. A finite-state approach to event semantics. In *Proceedings of the 9th International Symposium on Temporal Representation and Reasoning (TIME-02), Manchester*, pages 124–131. IEEE Computer Society Press, July.

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

Benjamin Han 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.

L[auri] Karttunen, J[ean]-P[ierre] Chanod, G[regory] Grefenstette, and A[nne] Schiller. 1996. Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328, December.

Lauri Karttunen, Tamás Gaál, and André Kempe. 1997. Xerox finite-state tool. Technical report, Xerox Research Centre Europe, Grenoble, France, June. http://www.xrce.xerox.com/competencies/content-analysis/fssoft/docs/fst-97/xfst97.html.

Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. Number 65 in Research Monographs. M.I.T. Press, Cambridge, Massachusetts.

Jyrki Niemi. 2004. Kalenteriajanilmausten semantiikka ja generointi: semantiikan mallintaminen laajennettuina säännöllisinä lausekkeina ja lausekkeiden luonnolliskielisten vastineiden XSLT-pohjainen generointi [The semantics and generation of calendar expressions: Modelling the semantics as extended regular expressions and generating the corresponding natural-language expressions using XSLT]. Master's thesis, University of Helsinki, Department of General Linguistics, Helsinki, November.

Hans Jürgen Ohlbach and Dov Gabbay. 1998. Calendar logic. *Journal of Applied Non-classical Logics*, 8(4):291–324.

Hans Jürgen Ohlbach. 2004. Relations between fuzzy time intervals. In *Proc. 11th International Symposium on Temporal Representation and Reasoning (TIME 2004)*, pages 44–50.

Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Artificial Intelligence. Springer, Berlin.