Effective Representation and Efficient Management of Indeterminate Dates

Fabio Grandi
C.S.I.TE.-C.N.R. and D.E.I.S.
University of Bologna, Italy
fgrandi@deis.unibo.it

Federica Mandreoli
D.S.I.
University of Modena and Reggio Emilia, Italy
fmandreoli@dsi.unimo.it

Abstract

Management of indeterminate *temporal expressions is useful in a wide range of applications, from designing and querying temporal databases to knowledge representation and reasoning in artificial intelligence. In this paper, we focus on the representation and management of indeterminate* dates*, corresponding to a common use of temporal indeterminacy which can be found in (historical) texts written in natural language, as in expressions like: around 1624, near the end of the fourteenth century, etc. In this context, we adapt and improve the probabilistic approach designed for the TSQL2 language and further developed by Dyreson and Snodgrass, and show how it can be effectively and efficiently adopted for the management of indeterminate dates.*

1 Introduction

Management of temporally indeterminate expressions [10] is useful in a wide range of applications, from designing and querying temporal databases to knowledge representation and reasoning in artificial intelligence. For example, its relevance was acknowledged during the design of the consensual temporal query language TSQL2 and indeterminacy support was included in the language features [13]. The TSQL2 approach was further developed by Dyreson and Snodgrass in [3], where implementation lines and a complete discussion and comparison with previous and alternative approaches to temporal indeterminacy can also be found.

The present research stems from the computer management, in a Cultural Heritage environment, of historical text sources, which usually contain a lot of written indeterminate temporal expressions (like around 1624, near the end of the fourteenth century, etc.). In particular, our main interest in indeterminacy derives from our involvement in a project, which is being carried on at the University of Florence [11], aimed at publishing on the Web an XML-based electronic edition of the "*Historical-geographical dictionary of Tuscany*" by Emanuele Repetti, which is an encyclopaedic collection of information concerning Tuscany published in eight volumes between 1833 and 1846. *Repetti*'s Dictionary is composed of several hundreds of alphabetically ordered items, concerning notable places in Tuscany (from large towns to small villages, providing historical, archaeological and artistic information), physical land attributes (viz. mountains, rivers, lakes, wetlands, etc.) and special items (like statistical tables). The digital edition should incorporate hyper-textual links, also from items to additional multimedia data (e.g. pictures, maps) and should be made available on the Web to be worldwide accessible through the Internet. This aspect has a noteworthy relevance from a Cultural Heritage and also scientific point of view, as it frequently happens in medieval archaeology that written sources have the same importance as material evidence. It has already been pointed out how the role of Internet in archaeological investigation is continuously increasing, as wide, fast and easy sharing of information on the Web has a substantial impact on the archaeological methodology [2, 9], which could be ever boosted by the deployment of XML-related technologies (as also evidenced in [11, 12]).

In this framework, an outstanding and appealing feature of XML is the capability of easily encoding semantic information in digital documents as *meta-data* to be automatically used by advanced computer tools, like "intelligent" search engines. In the "XML/Repetti" project [6], we aim at extending our previous work on incorporating temporal semantics and temporal search facilities into the Web [8, 5, 4] using XML-related technologies. The main required extension is properly the object of this work: that is the development of an adequate support to temporal indeterminacy which is abundantly used in natural languages and, especially, in historical text sources.

2 Temporal Indeterminacy in Natural Language

When reading texts written in natural language, it is very common to find vague and imprecise temporal expressions concerning the validity of historical facts. In this context, the reference base unit on the time axis (that is the bottom granularity [1]) is the *day* and, thus, we are mainly concerned with the representation and management of *dates*. Therefore, the validity of a historical fact can be either instantaneous, for an event occurred on a particular date or represented by a time *interval* (i.e. a set of contiguous days) for facts with a non-null duration. In any case, the validity is often defined through complex descriptions, also involving multiple calendars and granularities, which are not easy to formalize in a semantically correct and useful way. However, as it will become clear in the rest of the paper, the presence of different calendars and granularities does not increase technical difficulties, as all the calendars in use have the day as bottom granularity and seemingly use the same lattice (in practice, the only granularities of interest are always: day, month, year and century), so that suitable conversion functions can easily be provided.

Starting form the analysis of a large *corpus* of historical sources as *Repetti*'s Dictionary, we classified the temporal expressions denoting indeterminate events into four main categories[1]. If we denote by the term Reference Temporal Expression (RTE) the time literal written in text, the four categories correspond to the use of temporal expressions with the form: "in RTE" (to reference a validity shorter than the RTE duration) for category C_1, "at the beginning (end) of RTE" for C_2 (C_3), "around RTE" for C_4, as in the following examples:

- The abbey was consecrated to St. Martin **in 1276**. (C_1)
- The third circuit of the city walls was added **at the beginning of the fourteenth century**. (C_2)
- The famous painter died from the plague **near the end of March 1532**. (C_3)
- The delegation of the Emperor arrived in Rome **around Christmas 1467**. (C_4)

Notice that in the (actually very frequent) C_1 case, we are in the presence of a so-called *granularity mismatch* [3], where a determinate expression with higher granularity is used to denote an indeterminate expression with lower granularity. As a matter of fact, it is quite likely that the example refers to an event, happened on a certain date located in 1276, rather than to an activity lasting for the whole year. Moreover, we cannot *a priori* ascertain on which day the event actually happened and there is no reason to prefer one date with respect to another. On the other hand, the example "The castle was restored after the fire **between 1549 and 1553**" concerns a real interval since the restoration action likely required several years to be completed. However, since it is not known the exact date the works began and ended, the expression denotes an indeterminate interval, whose boundaries are indeterminate C_1-type dates.

Summing up, every indeterminate temporal expression found in the text can be reduced to to an indeterminate date or to an interval whose boundaries are indeterminate dates falling in one of the categories above, whose representation is addressed in the next Section.

2.1 Representation of Indeterminate Dates

In the field of temporal databases, the mainstream solution for the management of temporal indeterminacy is the *probabilistic approach* introduced for the TSQL2 language [13] and further developed by Dyreson and Snodgrass [3]. In this approach, the occurrence of an event is represented as a random variable with a given probability distribution. We follow this approach for two reasons: it is appropriate, from a semantic point if view, to represent the validity of historical facts (we agree with the discussion and comparison with alternative methods in [3]) and seems amenable to efficient implementation.

In the probabilistic model, an indeterminate event t is represented through its *probability distribution* P, different from zero only in an interval of possible occurrence, whose boundaries (t^- and t^+) are said *lower support* and *upper support*:

$$t = (t^- \sim t^+, P) \qquad \text{where } P(i) = \Pr[t = i] \text{ with } \sum_{i=t^-}^{t^+} P(i) = 1, \text{ and } P($$

For query evaluation, two indeterminate instants are considered equivalent ($t_1 \equiv t_2$) iff they have exactly the same supports and distributions. Moreover, TSQL2 introduces a suitable extension of the temporal order relation, that is a new definition of the "*Before()*" primitive which is used to define all the other temporal comparison operators [13]. In the indeterminate semantics, the "*Before()*" primitive includes an additional parameter p to specify an ordering *plausibility*, whose value can range from 0 to 100 (high plausibility means high precedence probability between the compared instants). Its complete definition becomes thus:

$$\textit{Before}(p, t_1, t_2) := \neg(t_1 \equiv t_2) \wedge \Pr[t_1 < t_2] \geq p/100$$

where the precedence probability is evaluated as (P_k is the

[1] Actually, our classification is based on the analysis of texts written in Italian, but we think it can be applied to texts written in other languages as well.

distribution of t_k):

$$\Pr[t_1 < t_2] = \sum_{i<j} P_1(i)P_2(j) \tag{1}$$

It should be noted that, since the *Before()* evaluation is based on an hypothesis of statistical independence between the occurrences of t_1 and t_2, the use of "*Before()*" does not lend itself to a correct probabilistic evaluation of the ordering $t_1 < t_2 < t_3$ as "*Before*$(p, t_1, t_2) \wedge$ *Before*(p, t_2, t_3)", unless the support intervals of t_1 and t_3 are disjoint.

In [3], probability distributions are represented via approximated mass functions. The proposed method allows to store any possible distribution, whose probability mass is quantized into P equal "rods", into a (pruned) binary tree with a number of leaves between P and $2P$. Then, the precedence probability (1) is evaluated by means of an algorithm based on rod counting, which traverses one of the trees associated to t_1 and t_2 in a breadth-first fashion by means of a moving *pivot* and ends when enough rod pairs are counted to ensure, with the required plausibility, that (1) is true or the converse is false (early exit conditions). Although the algorithm does a lot of "virtual" comparisons in the first steps (half of the rod pairs are counted by the first pivot), it slows down in an exponential way and, in the worst case, it has to count all the pairs and, thus, $2P$ pivots are required. Since for each pivot the algorithm traverses a subtree to locate a leaf (for both t_1 and t_2 mass-trees), the worst-case performance is $O(P \log_2 P)$. The P value cannot be too small (a trade-off value of 2^8 is used in [3]), because the approximation error, either in the probability mass quantization and in the algorithm counting, is $O(1/P)$. The space occupation of each mass-tree is between about $2P \log_2 PC$ (with best pruning) and $4P \log_2 PC$, where C (coarseness) is the number of sample points chosen for the discretization (with the value 2^{16} used in [3], one mass-tree requires between 1.5KB and 3KB). The total space required to represent a collection of indeterminate dates grows *linearly* with the size of the collection, as every date usually requires a different mass-tree (e.g. even two dates with distributions identical in shape and width but shifted on the date axis require two trees with the same rod counts in the leaves but different sample points stored in interior nodes). Thus, the representation of 10,000 indeterminate dates may require up to about 200MB to store the mass-trees, with P and C values as above. For our purposes, this framework is even too general (since it allows to represent any possible distribution, with the proviso that P is large enough) but quite costly in terms of time and space and, thus, we will introduce an alternative approach, based on piecewise-constant distributions, which is either appropriate to correctly represent all the required kinds of indeterminacy and to attain inexpensive storage and efficient query processing.

As far as the possible probability distributions we are interested in, we introduce here an assignment to the temporal expressions classified in the previous Section. The shapes we chose for the distributions are the simplest ones compatible with the "natural" meaning of the four categories, that is: flat for C_1, single-peaked positively (negatively) skewed for C_2 (C_3), single-peaked symmetric for C_4. The corresponding probability distributions are piecewise-constant over N intervals with the same amplitude (*base intervals*). We define *principal interval* the base interval associated with the maximum probability. For categories C_1 and C_4, the principal interval exactly corresponds to the RTE written in the text. For C_2 (C_3), the principal interval is the first (last) interval contained in the RTE at the immediately lower granularity level (e.g. the principal interval is January 1630 for the expression "at the beginning of 1630", whose RTE is 1630). The single values assigned to the probability over the base intervals have been computed by means of an *associated* continuous density. In the absence of more reasonable assumptions, we chose as associated densities the simplest ones with the same shape as our distributions[2]. Table I summarizes the results; for the distributions but the uniform, we also consider variants consisting in a greater or lesser dispersion around the mean value (e.g. VERY_LATE, WIDELY_AROUND, etc.), which will correspondingly imply a different number of base intervals.

Category	Shape	Associated density	Name
C_1	flat	uniform	DURING
C_2	positively skewed	exponential	VERY_EARLY
			EARLY
C_3	negatively skewed	reversed exponential	VERY_LATE
			LATE
C_4	symmetric	Gaussian	STRICTLY_AROUND
			AROUND
			WIDELY_AROUND

Table I. Probability distributions associated with indeterminate events.

Let us clarify the rationale of this encoding scheme with an example. Assume we have to understand and encode the expression "around year 1622" (C_4) and we can definitely exclude that the intended event could be happened before 1621 or after 1623 (N=3, STRICTLY_AROUND), while there is a certain probability that the event either happened in 1621 or in 1623 (with probability p' in both cases), though the probability p'' it happened right in 1622 (which is the principal interval) is maximum (obviously, $2p' + p'' = 1$). Hence, the probability distribution is (there are no leap

[2] More formally, we made the choices that maximize the entropy of the random variable when no additional information (e.g. moments) on the shape is available.

years between the supports):

$$P(i) = \begin{cases} 0 & \text{if } i < \text{1621/1/1 or } i > \text{1624/12/31} \\ p'/365 & \text{if 1621/1/1} \leq i \leq \text{1621/12/31 or 1623/1/1} \leq i \leq \text{1623/12/31} \\ p''/365 & \text{if 1622/1/1} \leq i \leq \text{1622/12/31} \end{cases}$$

The p' and p'' values are determined as mean values taken by the (continuous) associated Gaussian distribution on the same time intervals. The variance of the Gaussian is chosen such that 99,75% of the probability be contained between the supports (i.e. being negligible the contribution of the excluded tails). The resulting density functions are shown in Fig. 1 and we have $p' \simeq 15,8\%$ and $p'' \simeq 68,4\%$.

Notice that, with the mass-tree method and the parameter values used in [3], we could only represent our distributions with a rather *rough* approximation. For example, the probability mass over the base intervals of the `EARLY` distribution are, respectively, about 0.777, 0.179, 0.033 and 0.011; in order to quantize them with a reasonable approximation, we would need at least to exactly represent the third decimal digit, which requires $P = 2^{10}$. This would mean a five times more expensive comparison algorithm and a storage space between 6.5KB and 13KB for every mass-tree.

2.2 Encoding of Indeterminate Dates

With the proposed formalisms, every indeterminate date can be represented either by means of a support and distribution pair (as in TSQL2) or, equivalently, by means of a principal interval and distribution pair. While the first format is probably the best-suited to timestamp tuples in a valid-time relational database, we prefer the second as it is more appropriate to encode temporal information in a Cultural Heritage setting, like a repository of historical documents in digital form. In fact, it is more user-friendly for a non computer-expert user and allows to preserve in the encoding most of the semantic richness present in the original text expression. Therefore, we consider an indeterminate date encoding in the form: `(PRINCIPAL_INTERVAL, DISTRIBUTION)`, where the principal interval is expressed in *implicit* way, that is by means of its left boundary and its duration, both specified at a given granularity and according to a given calendar:

```
PRINCIPAL_INTERVAL = (START, GRANULARITY, DURATION, CALENDAR).
```

On the date axis, such an interval starts on the day given by *begin_of*(`START`) and is length is `DURATION` times the number of days in `GRANULARITY`, where the first day of `START` is determined according to the given `CALENDAR`. For instance, our sample expression "around year 1622" (with given distribution as in Fig. 1) can be simply encoded as:

```
((1622, YEAR, 1, GREGORIAN), STRICTLY_AROUND)
```

We would like to stress that the main reason of this encoding scheme is that the principal interval of the probability distribution has a direct correspondence with the RTE written in the text (1622 in our example). Otherwise, an explicit expression of the supports would require a detailed knowledge of the shape of the distributions and annoying computations *from the user*. In the example, the `STRICTLY_AROUND` distribution has three base intervals (i.e. "lobes" in Fig. 1) and, thus, the supports are $1621/1/1$ and $1623/12/31$ but, if we had defined it with five base intervals, they were $1620/1/1$ and $1624/12/31$. The availability of parameterized pre-defined distributions and the implicit support encoding scheme makes it a bit more "transparent" and user-friendly, so that the user can best concentrate on the choice of an intuitive "form factor" among a few available alternatives rather that on mathematical details of distributions like the support computation (or the variance). However, the lower and upper supports (and all the base intervals) can automatically be computed in a straightforward way from the principal interval and the specified distribution.

We emphasize that the uniform adoption of an encoding scheme, in which value and granularity have a direct correspondence with the RTE used in the text source (which we call **rigorous encoding rule**), represents itself *meta-information* (on the original form of the text contents) which can be used in automatic way for advanced investigations. Furthermore, different RTEs can be specified at different granularity levels and according to different calendars. In our framework, this fact only impacts on the correct conversion between the RTE and the corresponding principal interval of the distribution, for which simple rules can be provided. In fact, principal intervals are anchored on the date axis (at the base granularity of days) and all the other granularities can easily be converted to days, for each calendar commonly used in historical sources. Nevertheless, we maintain trace of the original form (including granularity and calendar) of the RTEs in their encoding as it represents potentially useful meta-information. For instance, the RTE "469 *ab urbe condita*" (i.e. 469 after the founding of Rome, which was in 753 B.C.) gives rise to the equivalent forms:

```
(0469-01-01, DAY, 365, ROMAN)
(-0284, YEAR, 1, GREGORIAN)
(0469, YEAR, 1, ROMAN)
```

where only the last one respects the rigorous encoding rule.

3 Efficient Comparison of Indeterminate Dates

We describe in this Section how the precedence probability between two indeterminate instants $t_1 = (t_1^- \sim t_1^+, P_1)$ and $t_2 = (t_2^- \sim t_2^+, P_2)$ can be evaluated in an efficient way. Starting from the definition (1), we will derive a formula that exploits the fact that distributions as in Tab. I are piecewise-constant functions over N base intervals, with equal amplitude $h = (t^+ - t^-)/N$, defined as:

$$I^k = [t_k^-, t_k^+) \qquad \text{where } t_k^- = t^- + h(k-1)\ , \quad t_k^+ = t^- + h\,k$$

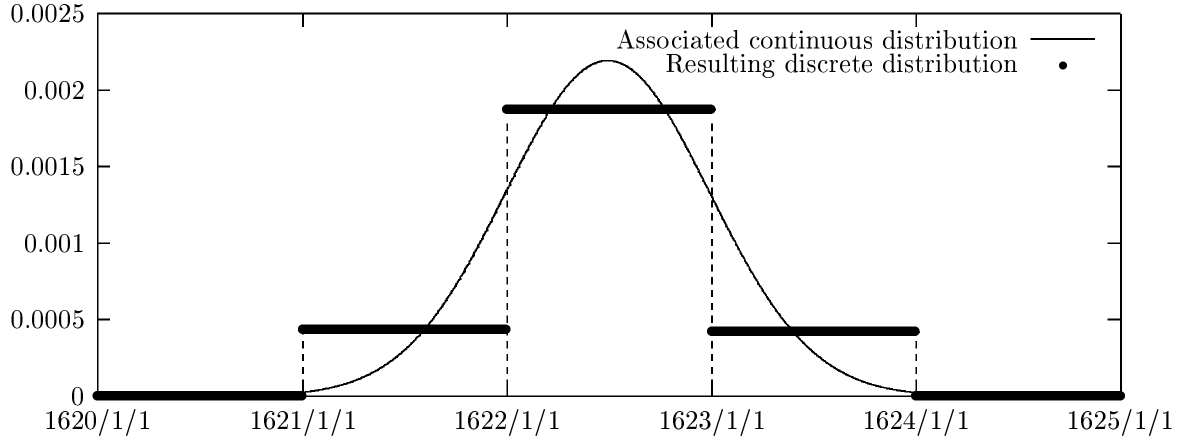

Figure 1. The `STRICTLY_AROUND` **distribution.**

As far as probabilities are concerned, if we write as p^k/h the constant occurrence probability $P(t)$ of each instant t in the base interval I^k, we have $\Pr[t \in I^k] = \Pr[t_k^- \leq t < t_k^+] = p^k$ (obviously $\sum_{k=1}^{N} p^k = 1$). For convenience, we define the function $I(x) = \lfloor (x - t^-)/h + 1 \rfloor$ which gives the index number k of the base interval I^k containing x.

We now evaluate the precedence probability between t_1 and t_2 (obviously $\Pr[t_1 < t_2]$ is 1 if $t_1^+ \leq t_2^-$ and 0 if $t_2^+ \leq t_1^-$). From the definition (1) we have:

$$\Pr[t_1 < t_2] = \sum_{i=-\infty}^{\infty} P_1(i) \cdot \sum_{j=i+1}^{\infty} P_2(j) = \sum_{i=t_1^-}^{t_1^+} P_1(i) \cdot \sum_{j=i+1}^{t_2^+} P_2(j) \tag{2}$$

where $\sum_{j=i+1}^{t_2^+} P_2(j) = \Pr[t_2 > i]$ measures the probability that the event t_2 occurs after the instant i. Therefore, the i-th term in the external summation computes the probability that t_1 occurs at i and t_2 occurs after. In order to evaluate (2), we will partition the range of the external summation (i.e. the t_1 support) into portions defined by the base intervals of t_2 (i.e. we "project" on the t_1 axis the partition defined on the t_2 axis) as follows:

$$\Pr[t_1 < t_2] = \sum_{i=t_1^-}^{t_2^- -1} P_1(i) \cdot \sum_{j=i+1}^{t_2^+} P_2(j) + \sum_{k=1}^{N_2} \sum_{i \in I_2^k} P_1(i) \cdot \sum_{j=i+1}^{t_2^+} P_2(j) \tag{3}$$

The first summation accounts (when $t_1^- \leq t_2^-$) for the left portion of the t_1 support possibly not overlapped by the t_2 support. Moreover, the inner summation is simply 1, as always $i + 1 \leq t_2^-$ (the whole t_2 distribution is to the right). In each of the portions defined by the t_2 base points in the second summation, say in the k-th which is actually I_2^k, the contribution of the innermost summation can be split into two parts, the former due to the P_2 probabilities inside I_2^k and the latter due to the P_2 probabilities outside (i.e. to the right of I_2^k). This yields:

$$\Pr[t_1 < t_2] = \sum_{i=t_1^-}^{t_2^- -1} P_1(i) + \sum_{k=1}^{N_2} \sum_{i=t_{2,k}^-}^{t_{2,k}^+ -1} P_1(i) \left(\sum_{j=i+1}^{t_{2,k}^+ -1} P_2(j) + \sum_{j=t_{2,k}^+}^{t_2^+} P_2(j) \right) \tag{4}$$

Now it can be shown that we do not make a significant error[3] if we substitute the sum $\sum_{j=i+1}^{t_{2,k}^+ -1} P_2(j)$ with the mean value $p_2^k/2$ (the summand is constant as $P_2(j)$ yields p_2^k/h_2 in I_2^k). Hence, we can introduce a function $G(k) = p_2^k/2 + \sum_{j=k+1}^{N_2} p_2^j = \Pr[t \geq t_{2,k}^- + h_2/2]$ to substitute the last parenthesis and obtain (we also split the first summation into a first one spanning only complete I_1^ℓ intervals and one over the left portion of the interval containing t_2^-, which is $I_1(t_2^-)$):

$$\Pr[t_1 < t_2] = \sum_{i=t_1^-}^{t_{1,I_1(t_2^-)}^- -1} P_1(i) + \sum_{i=t_{1,I_1(t_2^-)}^-}^{t_2^- -1} P_1(i) + \sum_{k=1}^{N_2} G(k) \sum_{i=t_{2,k}^-}^{t_{2,k}^+ -1} P_1(i) \tag{5}$$

If t_1 is the event with the largest base interval (i.e. $h_1 \geq h_2$), each I_2^k interval can only overlap one or two I_1^ℓ intervals. In the former case, I_2^k is completely contained in the I_1^ℓ interval with $\ell = I_1(t_{2,k}^+) = I_1(t_{2,k}^-)$, the inner sum in (5) has only one term:

$$\frac{t_{2,k}^+ - t_{2,k}^-}{h_1} p_1^{I_1(t_{2,k}^-)} = \frac{h_2}{h_1} p_1^{I_1(t_{2,k}^-)}$$

In the latter case, I_2^k contains the $t_{1,\ell}^+ = t_{1,\ell+1}^-$ base point with $\ell = I_1(t_{2,k}^-)$ (this fact can be detected by $I_1(t_{2,k}^+) -$

[3] For example, when I_2^k only contains one I_1^ℓ interval, this means to replace the exact value $\frac{h_2-1}{2h_1} p_2^k p_1^{I_1(t_{2,k}^-)}$ of $\sum_{i,j \in I_2^k; i<j} P_1(i) P_2(j)$ with the approximate value $\frac{h_2}{2h_1} p_2^k p_1^{I_1(t_{2,k}^-)}$.

$I_1(t^-_{2,k}) = 1$), the inner sum in (5) has two terms:

$$\frac{t^+_{1,I_1(t^-_{2,k})} - t^-_{2,k}}{h_1} p_1^{I_1(t^-_{2,k})} + \frac{t^+_{2,k} - t^-_{1,I_1(t^+_{2,k})}}{h_1} p_1^{I_1(t^+_{2,k})}$$

If we introduce the function $f(x) = \frac{x - t^-_{I(x)}}{h} p^{I(x)}$ (representing the occurrence probability over a fraction of a base interval from the beginning to x) we can simply rewrite the precedence probability as:

$$\Pr[t_1 < t_2] = \sum_{\ell=1}^{I_1(t_2^-)-1} p_1^\ell + f(t_2^-) + \qquad (6)$$
$$\sum_{k=1}^{N_2} G(k) \left[\frac{h_2}{h_1} p_1^{I_1(t^-_{2,k})} + \left(I_1(t^+_{2,k}) - I_1(t^-_{2,k}) \right) \left(\left(1 - \frac{h_2}{h_1} \right) p_1^{I_1(t^-_{2,k})} + f(t^+_{2,k}) - f(t^-_{2,k}) \right) \right]$$

As a final remark, we can notice that if the t_1 support ends before the t_2 support (i.e. $t_1^+ < t_2^-$), formula (6) is still valid if we extend the definition of the t_1 base intervals also for $\ell > N_1$ (obviously with $p_1^\ell = 0$). However, in this case, the sum can be actually stopped in the last I_2^k interval overlapping the t_1 support (i.e. when $k = I_2(t_1^+)$).

When the events to be compared are such that $h_1 < h_2$, we can use the formula $\Pr[t_1 < t_2] = 1 - \Pr[t_2 < t_1] - \Pr[t_2 = t_1]$, where $\Pr[t_2 < t_1]$ can still be evaluated with formula (6) and, as it is easy to verify, $\Pr[t_1 = t_2]$ (if $h_1 > h_2$) can be expressed as:

$$\Pr[t_1 = t_2] = \sum_{k=1}^{N_2} p_2^k \left[\frac{h_2}{h_1} p_1^{I_1(t^-_{2,k})} + \left(I_1(t^+_{2,k}) - I_1(t^-_{2,k}) \right) \left(\left(1 - \frac{h_2}{h_1} \right) p_1^{I_1(t^-_{2,k})} + f(t^+_{2,k}) - f(t^-_{2,k}) \right) \right]$$

which can be also evaluated during the computation of (6) without an increase in computational complexity (the term in square brackets is the same).

Finally, notice that, by means of the cumulative probability function over base intervals $F(i) = \sum_{j=1}^{i} p^i$, we can also write:

$$\sum_{\ell=1}^{I_1(t_2^-)-1} p_1^\ell = F_1(I_1(t_2^-) - 1)\,, \qquad G(k) = \tfrac{1}{2} p_2^k + 1 - F_2(k)$$

Therefore, if we pre-compute all the $F(i)$ values (in $O(N)$ time) for every supported distribution and store them in a system table, then the formulas above can simply be evaluated via table-lookup and the complexity of (6) reduces to $O(N_2)$ (and evaluation of $G(k)$ in the summation is inexpensive).

3.1 Optimization of Temporal Selection

The precedence probability, as in the TSQL2 approach [13], can be used for the definition of the temporal selection predicates working on events and/or intervals to be practically used in the queries (e.g. *precedes*, *overlaps*, *contains* and *meets*). With indeterminate dates, they can be considered satisfied when the associated probability is greater than or equal to the assigned plausibility. For example, considering intervals $I_1 = [I_1^s, I_1^e]$ and $I_2 = [I_2^s, I_2^e]$, such probabilities (assuming independence between the interval boundaries) can be evaluated as:

$$\Pr[I_1 \textit{ precedes } I_2] = \Pr[I_1^e < I_2^s]\,, \qquad \Pr[I_1 \textit{ overlaps } I_2] = \mathrm{F}$$
$$\Pr[I_1 \textit{ contains } I_2] = \Pr[I_1^s < I_2^s] \cdot \Pr[I_1^e < I_2^e]\,, \qquad \Pr[I_1 \textit{ meets } I_2$$

Using the formulas obtained in the previous Section, each of these probabilities can be evaluated in $O(N)$ steps, where N is the number of base points of the indeterminate date with the smallest base interval (i.e. it is N_2 if $h_1 > h_2$). In any case, since the only distributions of interest are those listed in Tab. 1, N is always a very limited number, never exceeding 7, hence we can assume a constant time to evaluate the precedence probability, as it does non depend on the problem dimension. Notice that also the space overhead required to store the system table for quick evaluation of the F function is negligible, as we need just to store a total of 30 values for all the distributions (each one contributes with its N), regardless of the number of different indeterminate dates we have to represent.

In the following, we consider a further optimization that can be used during the evaluation of a query, that is when a temporal predicate has to be matched against every tuple in a table, in a temporal relational database, or against every document in a collection, in an information retrieval setting. In order to cover both (and other) cases, in the sequel we will talk about a generic search engine looking for qualifying items. In particular, the proposed algorithm exploits the fact that several comparisons can actually be avoided during query processing, thanks to the results of the previously effected comparisons. In order to exemplify this optimization, consider the following query: *Search the repository for all the items containing dates following t_Q (at plausibility level p).* In the most general case, the query input is also an indeterminate event $t_Q = (t_Q^- \sim t_Q^+, P_Q)$. In order to compute the answer, the search engine has to scan all the items to test if any date t_D they contain satisfies the *Before(p, t_Q, t_D)* predicate. The further optimization is based on the maintenance, for each distribution form P, of two bounds:

- Upper bound $UB(P)$: defined as the maximum known upper support t_D^+ of a date t_D with distribution P which *surely* is after the query date t_Q (at plausibility level p);

- Lower bound $LB(P)$: defined as the minimum known lower support t_D^- of a date t_D with distribution P which *surely* is not after the query date t_Q (at plausibility level p).

The two bounds are initialized as $UB(P) = t_Q^-$, $LB(P) = t_Q^+$ and, during the query execution, the processing of the date $t_D = (t_D^- \sim t_D^+, P_D)$ is effected as follows. If $t_D^+ \leq UB(P_D)$ or $t_D^- \geq LB(P_D)$, there is no need to evaluate the precedence probability: in the former case, we are sure that *Before*(p, t_Q, t_D) is true and the date t_D qualifies for the query; in the latter, we are sure that *Before*(p, t_Q, t_D) is false and the date t_D can be discarded. Otherwise, $\wp = \Pr[t_Q < t_D]$ is evaluated: if $\wp \geq p/100$, the date t_D qualifies and the upper bound is updated to the new value $UB(P_D) = t_D^+$, else the date t_D can be discarded and the lower bound is updated to the value $LB(P_D) = t_D^+$. In this way, many useless comparisons are actually avoided (according to the results of previous comparisons involving dates with the same distribution). Every time a comparison has to be effected, the result is also used to tighten the bounds, making gradually narrower the date range for which the precedence probability evaluation has to be done.

4 Conclusions

In this paper we presented a probabilistic approach for the representation and management of indeterminate dates in historical text sources in digital form. The approach is based on the use of piecewise-constant distributions, which is fairy correct from a semantic viewpoint and computationally very efficient, as it is provided with optimized comparison algorithms and does not require storage space overhead to represent probability distributions. In this respect, our encoding scheme is also very attractive for use in very large temporal databases, provided that piecewise-constant distributions are considered suited as well for an effective modeling of indeterminacy in the given application context.

References

[1] C. Bettini, C.E. Dyreson, W.S. Evans, R.T. Snodgrass, X. Sean Wang, "A Glossary of Time Granularity Concepts", in *Temporal Databases - Research and practice*, LNCS N. 1399, Springer-Verlag, 1998.

[2] I. Bogdanovic, O. Vicente, J.A. Barcelo, "A Theory of Archaeological Knowledge Building by using Internet: the DIASPORA Project", Proc. of *Intl' Conf. on Computer Applications in Archaeology (CAA'99)*, Dublin, Ireland, 1999.

[3] C.E. Dyreson, R.T. Snodgrass, "Supporting Valid-time Indeterminacy", *ACM Trans. on Database Systems*, Vol. 23, No. 1, 1998.

[4] F. Grandi, F. Mandreoli, "The Valid Web ©", Proc. of *Software Demonstrations Track at the EDBT'2000 Intl. Conf.*, Konstanz, Germany, March 2000.

[5] F. Grandi, F. Mandreoli, "The Valid Web: an XML/XSL Infrastructure for Temporal Management of Web Documents", Proc. *Intl' Conf. on Advances in Information Systems (ADVIS'2000)*, Izmir, Turkey, 2000, LNCS N. 1909, Springer-Verlag, 2000.

[6] F. Grandi, F. Mandreoli, "The "XML/Repetti" Project: XML Encoding and Manipulation of Temporal Information in Historical Text Sources", *submitted for publication.*

[7] F. Grandi, F. Niccolucci, "XML Technologies for the Representation and Management of Spatiotemporal Information in Archaeology", Proc. of *Intl' CODATA Conference*, Baveno, Italy (abstract available in *CODATA 2000 Book of Abstracts*, CODATA Secretariat, Paris, 2000).

[8] F. Grandi, M. R. Scalas, "Extending Temporal Database Concepts to the World Wide Web", Proc. of *Natl' Conf. on Advanced Database Systems (SEBD'98)*, Ancona, Italy, 1998.

[9] S. Hermon, F. Niccolucci, "The Impact of Web-shared Knowledge on Archaeological Scientific Research", Proc. of *Intl' Conf. on Current Research on Information Systems (CRIS2000)*, Helsinki, Finland, 2000.

[10] C.S. Jensen, J. Clifford, R. Elmasri, S.K. Gadia, P. Hayes, S. Jajodia (eds.) *et al.*, "A Consensus Glossary of Temporal Database Concepts - February 1998 Version", in *Temporal Databases - Research and practice*, LNCS N. 1399, Springer-Verlag, 1998.

[11] F. Niccolucci, A. Zorzi, M. Baldi, F. Carminati, P. Salvatori, T. Zoppi, "Historical Text Encoding: an Experiment with XML on Repetti's Historical Dictionary", Proc. of *Conf. of the Association for History and Computing - UK Branch (AHC-UK'99)*, London, UK, 1999.

[12] A. Benvenuti, F. Niccolucci, S. Baragli, C. Carpini, "Advances in XML Treatment of Historical Documents", in *La Historia en una Nueva Frontera*, AHC, Toledo, Spain, 2000.

[13] R.T. Snodgrass (ed.), I. Ahn, G. Ariav, D. Batory, J. Clifford, C.E. Dyreson, R. Elmasri, F. Grandi, C.S. Jensen, W. Käfer, N. Kline, K. Kulkarni, T.Y. Cliff Leung, N. Lorentzos, R. Ramakrishnan, J.F. Roddick, A. Segev, M.D. Soo, S.M. Sripada, *The TSQL2 Temporal Query Language*, Kluwer Academic Publishers, Boston, MA, 1995.