A Constraint-based Modeling of Calendars

Benjamin Han
IBM Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598, U.S.A.
dbhan@us.ibm.com

Abstract

Temporal information in the real world is often presented in an incomplete fashion. In this paper we propose a constraint-based calendar system suitable for inferring implicit calendric information. Our approach views calendars as constraint satisfaction problems, and time points are represented as instantiations of temporal units. We also present a semantic model for the system to justify the imposed design requirements, and describe selected services provided by such systems.

Introduction

Consider the following news story written in 2005 regarding the most recent space shuttle accident:

> *Over the long holiday weekend honoring* ***Martin Luther King Jr.'s birthday****, the mission managers suspended their meetings... They met* ***on the next day the 21st****...*

For a system to deduce that the meeting took place on January 21, 2003, it needs to know that Martin Luther King Jr.'s birthday is a Monday in January, and in this particular scenario its next day, Tuesday, is January 21. With these requirements the system can then infer that the possible years are 2003, 1997, 1992, etc, with 2003 being the most recent one. This requires, among other things, a calendar system that captures the relations among different temporal units (year, month, etc) to make reasoning with incomplete information possible.

In this paper we present a constraint-based model of calendars, which has primarily been motivated by the development of natural language applications. Contrasting to the previous proposals of modeling calendars as an algebraic system (Bettini and Sibi 2000; Peng Ning, Wang, and Jajodia 2002), as a logical framework (Ohlbach and Gabbay 1998; Combi, Franceschet, and Peron 2004), or as a string-based representation (Wijsen 2000), one major characteristic of our approach is that we view a calendar as a *constraint system*: different temporal units are related via constraints (e.g., February in a non-leap year cannot have 29 days), and more information can be inferred from a usually under-specified natural language expression. Our calendar model is also designed to work with a higher-level temporal representation, Time Calculus for Natural Language (Han, Gates, and Levin 2006), to capture semantics of expressions such as "*the third Monday in January this year*" (written as $\{|3_{\{\mathrm{mon}\}}|@\{\mathtt{jan}, \{\mathtt{now} + |0_{\mathrm{year}}|\}\}\}$ in TCNL). We should emphasize, however, that our approach is not mutually exclusive to the other proposals mentioned earlier – it remains to be explored if the constraints among temporal units can be implemented in an algebraic or string-based fashion.

The rest of this paper is organized as follows. First in the next section we describe the basic structure of our calendar constraint system, the concept of a time point, and the solution methods of the system. The vanilla constraint system is then souped up with two relations in the following section, with which concepts such as granularity and anchoring status are then defined. In the next section we describe complete calendars built by multiple calendar components and the requirements for them. We then describe how some of the calendar services can be provided. Finally we conclude the paper with a summary.

Calendar as a Constraint System

At the most basic level a calendar contains two kinds of entities: *temporal units* and *temporal values*. Each unit can be viewed as a variable that takes on a set of fully-ordered values; e.g., unit `year` can be assigned with value 2006, and unit `month` can be assigned with value "May", etc. This unit/value dichotomy enables us to treat a calendar as a constraint satisfaction problem (CSP) (Dechter 2003), defined as follows.

Definition 1. A *calendar CSP* is a triple $(\mathcal{U}, \mathcal{D}, \mathcal{R})$ where $\mathcal{U} = \{u_1, \ldots, u_n\}$ is a set of *temporal units* with respective domains $\mathcal{D} = \{\mathcal{D}_{u_1}, \ldots, \mathcal{D}_{u_n}\}$, and $\mathcal{R} = \{\mathcal{R}_{\mathcal{U}_1}, \ldots, \mathcal{R}_{\mathcal{U}_m}\}$ is a set of *constraints* among the units. Each $\mathcal{D}_{u_i} = (V_{u_i}, <_{V_{u_i}})$ specifies a fully-ordered set of values, and each $\mathcal{R}_{\mathcal{U}_j}$ is a relation defined over a subset of units $\mathcal{U}_j \subseteq \mathcal{U}$, where $\mathcal{U}_j$ is called the *scope* of $\mathcal{R}_{\mathcal{U}_j}$.

As an example, Fig. 1 shows a simplified Gregorian calendar CSP featuring units year, quarter of year (`qoy`), semester (`sem`), month and day. Note that we do not explicitly define $\mathcal{R}_{\{\mathrm{year},\mathrm{qoy}\}}$ and $\mathcal{R}_{\{\mathrm{year},\mathrm{month}\}}$ because every possible pair in the Cartesian products of the involved values is allowed. Also

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

$$
\begin{aligned}
\mathcal{U} &:= \{\texttt{year}, \texttt{qoy}, \texttt{sem}, \texttt{month}, \texttt{day}\} \\
\mathcal{D} &:= \{(V_{\text{year}}, <_{\mathbb{N}}), (V_{\text{qoy}}, <_{\mathbb{N}}), \\
&\quad (V_{\text{sem}}, \texttt{spring} < \texttt{summer} < \texttt{fall}) \\
&\quad (V_{\text{month}}, \texttt{jan} < \ldots < \texttt{dec}), (V_{\text{day}}, <_{\mathbb{N}})\} \\
\mathcal{R} &:= \{\mathcal{R}_{\{\text{qoy,month}\}}, \mathcal{R}_{\{\text{sem,month}\}}, \mathcal{R}_{\{\text{month,day}\}}, \mathcal{R}_{\{\text{year,month,day}\}}\} \\
V_{\text{year}} &:= \{1753, \ldots, 3000\} \\
V_{\text{qoy}} &:= \{1, \ldots, 4\} \\
V_{\text{sem}} &:= \{\texttt{spring}, \texttt{summer}, \texttt{fall}\} \\
V_{\text{month}} &:= \{\texttt{jan}, \ldots, \texttt{dec}\} \\
V_{\text{day}} &:= \{1, \ldots, 31\} \\
\mathcal{R}_{\{\text{qoy,month}\}} &:= (\{1\} \times \{\texttt{jan}, \texttt{feb}, \texttt{mar}\}) \\
&\quad \cup (\{2\} \times \{\texttt{apr}, \texttt{may}, \texttt{jun}\}) \\
&\quad \cup (\{3\} \times \{\texttt{jul}, \texttt{aug}, \texttt{sep}\}) \\
&\quad \cup (\{4\} \times \{\texttt{oct}, \texttt{nov}, \texttt{dec}\}) \\
\mathcal{R}_{\{\text{sem,month}\}} &:= (\{\texttt{spring}\} \times \{\texttt{jan}, \texttt{feb}, \texttt{mar}, \texttt{apr}, \texttt{may}\}) \\
&\quad \cup (\{\texttt{summer}\} \times \{\texttt{jun}, \texttt{jul}\}) \\
&\quad \cup (\{\texttt{fall}\} \times \{\texttt{aug}, \texttt{sep}, \texttt{oct}, \texttt{nov}, \texttt{dec}\}) \\
\mathcal{R}_{\{\text{month,day}\}} &:= (\{\texttt{jan}, \texttt{mar}, \texttt{may}, \texttt{jul}, \texttt{aug}, \texttt{oct}, \texttt{dec}\} \times V_{\text{day}}) \\
&\quad \cup (\{\texttt{apr}, \texttt{jun}, \texttt{sep}, \texttt{nov}\} \times \{1, \ldots, 30\}) \\
&\quad \cup \{(\texttt{feb}, 29)\} \\
\mathcal{R}_{\{\text{year,month,day}\}} &:= (V_{\text{year}} \times \{\texttt{jan}, \texttt{mar}, \ldots, \texttt{dec}\} \times V_{\text{day}}) \\
&\quad \cup (\mathit{leapYears} \times \{\texttt{feb}\} \times \{1, \ldots, 29\}) \\
&\quad \cup (\mathit{commonYears} \times \{\texttt{feb}\} \times \{1, \ldots, 28\}) \\
\mathit{leapYears} &:= \{y \in V_{\text{year}} \mid y \equiv 0 \bmod 4 \\
&\quad \text{and } (y \not\equiv 0 \bmod 100 \text{ or } y \equiv 0 \bmod 400)\} \\
\mathit{commonYears} &:= V_{\text{year}} \setminus \mathit{leapYears}
\end{aligned}
$$

Figure 1: A simplified Gregorian calendar

note that the constraint $\mathcal{R}_{\{\texttt{year,month,day}\}}$ alone permits invalid dates such as April 31, 2007, but they will be rejected by the constraint $\mathcal{R}_{\{\texttt{month,day}\}}$.

Related to a calendar CSP is the concept of a time point. A straightforward way of representing a point is to treat it as a set of assignments to the units of a calendar CSP, namely, an *instantiation* of those units; e.g., the time point "*February 29*" can be represented by the instantiation $\{\texttt{feb}_{\texttt{month}}, \texttt{29}_{\texttt{day}}\}$, which is incomplete because not every unit existing in the calendar is assigned[1]. Formally we define a *coordinate* in time as follows.

Definition 2. Given a calendar CSP $P = (\mathcal{U}, \mathcal{D}, \mathcal{R})$, a *coordinate* c with *scope* $\mathcal{S}(c) \subseteq \mathcal{U}$ is an instantiation of the units in $\mathcal{S}(c)$, and c is called a *complete coordinate* if $\mathcal{S}(c) = \mathcal{U}$. The *extensions* of c, written as $\mathcal{E}(c)$, is the set of all complete coordinates c' that are solutions to P and $c \subseteq c'$. Finally c is *consistent* if $\mathcal{E}(c)$ is not empty.

Intuitively speaking, extensions are all possible interpretations of a coordinate, therefore empty extensions indicate that there is no plausible interpretation for the coordinate. Using the example calendar CSP above we have $c = \{\texttt{feb}, \texttt{29}_{\texttt{day}}\}$ as a consistent coordinate since

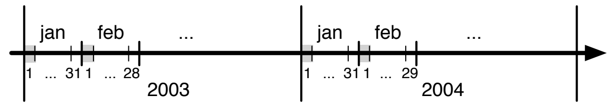

Figure 2: A timeline fragment

$\{\texttt{2008}_{\texttt{year}}, \texttt{feb}, \texttt{29}_{\texttt{day}}, \ldots\} \in \mathcal{E}(c)$ are possible interpretations, and $\{\texttt{apr}, \texttt{31}_{\texttt{day}}\}$ is inconsistent. If we want to obtain a set of possible interpretations over only a subset of units, we need to project the extensions on the set as described below.

Definition 3. Let X be a set of coordinates and $\mathcal{U}$ be a set of units. The *set projection* of X on $\mathcal{U}$ is defined as $\bar{\pi}_{\mathcal{U}}(X) = \{\pi_{\mathcal{U}}(x) \mid x \in X\}$.[2] When $X = \mathcal{E}(c)$ (the extensions of coordinate c) we call $\bar{\pi}_{\mathcal{U}}(\mathcal{E}(c))$, written as $\mathcal{E}_{\mathcal{U}}(c)$, the *extension projection* of c on $\mathcal{U}$.

For example given a coordinate $c = \{\texttt{2006}_{\texttt{year}}, \texttt{1}_{\texttt{qoy}}\}$ ("*the first quarter of 2006*") and $\mathcal{U} = \{\texttt{year}, \texttt{month}\}$, we have

$$\mathcal{E}_{\mathcal{U}}(c) = \{\{\texttt{2006}_{\texttt{year}}, \texttt{jan}\}, \{\texttt{2006}_{\texttt{year}}, \texttt{feb}\}, \{\texttt{2006}_{\texttt{year}}, \texttt{mar}\}\}.$$

Thus the application of an extension projection on a coordinate is akin to conducting a "time conversion", where an original denotation is expressed using a different set of temporal units.

Semantics of a Calendar CSP

A calendar CSP in Definition 1 is not required to conform to any reality. In fact it is possible to design constraints that let in "impossible" times such as *February 30* or *Monday, January 2, 2007*. In the following we will define the underlying conceptual model of a calendar CSP, and describe the requirements for a well-defined instance. Note that in operations our calendar systems do not directly manipulate this conceptual model. Our goal is replacing any direct manipulation of the *semantic* model with a *syntactic* operation performed on a calendar CSP.

Intuitively a value assignment to a temporal unit in a calendar CSP should be mapped to a set of intervals on a timeline – an integer line representing our conception of time. Consider a fragment of a timeline in Fig. 2. The value `2003` of the unit `year` is mapped to a single interval representing the year 2003, and the value `1` of the unit `day` is mapped to a set of intervals representing January 1, 2003, February 1, 2003, etc. Certain value assignments will be mapped to intervals of size zero if they are impossible times; for example, `30` of the unit `day` is mapped to a null interval when it is in February (omitted on the timeline in Fig. 2). This intuition is formalized by the *time mapping* function.

Definition 4. Let u be a temporal unit with values V_u. The *time mapping* of u is a set $T_u(v) = \{T_u^1(v), \ldots, T_u^p(v), \ldots\}$ where $T_u^p(v)$, the p-th *period* of v ($p \in \mathbb{N}$), maps $v \in V_u$ to a semi-open *integer* interval $[\mathit{start}_{v,p}, \mathit{end}_{v,p})$ with the following requirements:

1. $T_u^p(v_i)$ *meets* $T_u^p(v_{i+1})$ (v_i and v_{i+1} are two successive values in V_u); i.e., $\mathit{end}_{v_i,p} = \mathit{start}_{v_{i+1},p}$;

[1] In the rest of the paper we shall drop unit subscripts when writing a coordinate if no ambiguity is possible; e.g., $\{\texttt{feb}, \texttt{29}_{\texttt{day}}\}$.

[2] The projection of c is defined as $\pi_{\mathcal{U}_j}(c) = \{v_u \mid v_u \in c \text{ and } u \in \mathcal{U}_j\}$

2. $T_u^p(v_n)$ *meets* $T_u^{p+1}(v_1)$ (v_1/v_n is the minimal/maximal value in V_u); i.e., $end_{v_n,p} = start_{v_1,p+1}$.

Additionally we call $T_u^p(v)$ a *null period* if $start_{v,p} = end_{v,p}$.

Essentially the definition above captures the way a calendar is used to conceptualize time in a *cyclic* manner: within the finite set of values of a temporal unit, each successive value is mapped to a successive and adjacent interval, and upon reaching the maximal value we "restart" our mapping process from the minimal value again. In this process no stretch of the timeline is left unmapped.

The meaning of a temporal expression, on the other hand, must be built on the meaning of its parts. For example, the expression "*January 2006*" should denote the portion of the timeline where it is *both* in January and in the year 2006. Thus the time mapping of a coordinate is defined as the *intersection* of the time mappings of its value assignments.

Definition 5. Given two time mappings $T_{u_1}(v_1) = \{\ldots, t_i, \ldots\}$ and $T_{u_2}(v_2) = \{\ldots, t_j, \ldots\}$ (t_i and t_j are two periods), their intersection is defined as $T_{u_1}(v_1) \cap T_{u_2}(v_2) = \{\ldots, t_i \cap t_j, \ldots\}$ where $t_i \cap t_j \neq \emptyset$. The time mapping of a coordinate c is defined as $T(c) = \bigcap_{u \in \mathcal{S}(c)} T_u(\pi_u(c))$.

We can now define two ordering relations between any two time mappings: the chronological ordering ($<$) and the set inclusion ordering ($\subseteq$): if c_1 and c_2 be two coordinates, we say $T(c_1) \subseteq T(c_2)$ if for every interval $t_i \in T(c_1)$ there exists a $t_j \in T(c_2)$ such that $t_i \subseteq t_j$, and $T(c_1) < T(c_2)$ if the maximal integer in $T(c_1)$ is less than the minimal integer in $T(c_2)$, assuming both extremes exist. A simple corollary is that the superset relation between two coordinates becomes the subset relation between their time mappings.

Corollary 6. *If c_1 and c_2 are two coordinates and $c_1 \subseteq c_2$, then $T(c_2) \subseteq T(c_1)$.*

With the semantics of a coordinate defined in terms of its time mapping, we can also find the corresponding semantic interpretations for coordinate extensions and extension projections (Definition 1 and 3). We will however defer that discussion until after the introductions of the necessary concepts and results.

Using time mappings as our semantic model, we can now explicate the requirements of a *well-defined* calendar CSP as follows.

Definition 7. A calendar CSP P is *well-defined* if it implements a time mapping such that

1. every solution to P is mapped to a unique non-null period;
2. every non-null period must be mapped to by at least one consistent coordinate of P.

It turns out that with the help of Definition 4, the first requirement above establishes an *injection* mapping from the set of solutions to the set of non-null periods, namely two different solutions cannot occupy the same portion of the timeline.

Corollary 8. *Let P be a well-defined calendar CSP, c_1 and c_2 be two solutions to P and $c_1 \neq c_2$, then $T(c_1) \neq T(c_2)$.*

The simplified Gregorian calendar CSP presented earlier is a well-defined calendar CSP, *provided* the resolution of the underlying timeline is `day` (i.e., each integer denotes one day). This formulation therefore implies the existence of a "common-divider" unit among all units, because a coordinate is mapped to the intersection of the periods mapped from its individual value assignments. Without the help of this unit an intersection between two periods at different "granularity" cannot be guaranteed to yield an integral set of periods. We will later introduce the concept of granularity and describe how the existence of this common-divider unit is enforced.

In the rest of this paper we shall assume every calendar CSP is well-defined.

Solving Calendar CSPs

Conventional methods for constraint propagation and distribution can be used to solve a calendar CSP given a coordinate. In particular we use a standard AC-3 algorithm (Mackworth 1977) to eliminate illegal values from the domains of constrained temporal units, and a backtracking search procedure to iterate over the coordinates in an extension projection. Our implementation of the distribution algorithm can iterate forward or backward over $\mathcal{E}_{\mathcal{U}}(c)$ starting from a designated coordinate c_0. The overall time complexity for finding the first solution is $O(|\mathcal{R}| \cdot w^{|\mathcal{U}|})$ where w is the maximum size of any domain – this is a manageable cost since a typical calendar CSP is usually small.

Relational Structures

Although a calendar CSP captures the intersective nature of temporal units via its constraints (e.g., the constraint $\mathcal{R}_{\{\mathtt{month,day}\}}$ allows $\{\mathtt{feb}, \mathtt{28}_{\mathtt{day}}\}$ because $T_{\mathtt{month}}(\mathtt{feb}) \cap T_{\mathtt{day}}(\mathtt{28}) \neq \emptyset$), there are other characteristics of a calendar that are not captured by this basic formulation. The most pronounced one is one of a *hierarchical* nature, e.g., every year is composed of a set of consecutive quarters, which is then composed of a set of consecutive months, etc. Related to this property is the concept of *granularity*, namely how precise in time a particular coordinate is. A second characteristic is one of a *cyclic* nature, e.g., quarters are periodic in every year, but months are not periodic in every quarter. Explicating this property will allow us to distinguish between *anchored* coordinates vs. unanchored ones (e.g., $\{\mathtt{2006}_{\mathtt{year}}, \mathtt{feb}\}$ is anchored but $\{\mathtt{feb}\}$ is not), and only anchored coordinates can be compared chronologically to one another. In this section we will therefore propose two relations to capture these characteristics.

Measurement Relation

In a calendar a certain unit of time can always be measured by another unit, unless the unit is already at the finest possible resolution. Using the semantic model established earlier, we can capture this hierarchical nature of calendars in the *measurement* relation defined below.

Definition 9. Let u_1 and u_2 be two temporal units with values V_{u_1} and V_{u_2}, respectively. We say u_1 *is measured by* u_2, or $u_2 \dot{\leq} u_1$, if every *non-null* period $T_{u_1}^p(v_{u_1})$ ($v_{u_1} \in V_{u_1}$ and $p \in \mathbb{N}$) is a concatenation of *consecutive* value periods of unit u_2, namely

$$T^p_{u_1}(v_{u_1}) = \bigcup_{q \in Q \subset \mathbb{N}} \bigcup_{v_i \in V_q \subseteq V_{u_2}} T^q_{u_2}(v_i)$$

where $V_{u_2} = \{v_1, \ldots, v_n\}$, $1 \leq i \leq n$, and the set $\{(q-1) \cdot n + i\}$ is an interval of $\mathbb{N}$.

If $|Q| = |V_q| = 1$ (or $T^p_{u_1}(v_{u_1}) = T^q_{u_2}(v_{u_2})$) we say u_1 and u_2 are *measurement equivalent*, or $u_2 \doteq u_1$; otherwise $u_2 \dot{<} u_1$. If $|Q| = 1$ and $|V_q| > 1$ we also say u_1 is *immediately measured by* u_2, written as $u_2 \dot{\prec} u_1$.

Additionally we call u a *maximal unit* if $u' \dot{\leq} u$ for every unit u', and u must have only one period (its values never repeat on the timeline), i.e., $T_u(v) = \{T^1_u(v)\}$ for $v \in V_u$. We say u is a *minimal unit* if $u \dot{\leq} u'$ for every unit u', and define $\min(\cdot)$ as a function returning the set of minimal units in the argument set.

Note that the difference between $u_2 \dot{<} u_1$ and $u_2 \dot{\prec} u_1$ (u_1 is immediately measured by u_2) is that the latter requires *every* period of a value v_{u_1} to be composed of the time mappings of consecutive values of u_2 in the same q-th period. For example, $\texttt{month} \dot{\prec} \texttt{year}$ because $T^1_{\texttt{year}}(y) = T^q_{\texttt{month}}(\texttt{jan}) \cup \ldots \cup T^q_{\texttt{month}}(\texttt{dec})$ for every $y \in \mathbb{N}$, but $\texttt{day} \not\dot{\prec} \texttt{year}$ because $T^1_{\texttt{year}}(y) = T^q_{\texttt{day}}(1) \cup \ldots \cup T^q_{\texttt{day}}(\texttt{31}) \cup T^{q+1}_{\texttt{day}}(1) \cup \ldots \cup T^{q+1}_{\texttt{day}}(\texttt{29}) \cup \ldots$ (every `year` has more than one instance of $1_{\texttt{day}}$, $2_{\texttt{day}}$, etc); in contrast both $\texttt{month} \dot{<} \texttt{year}$ and $\texttt{day} \dot{<} \texttt{year}$ are true. The relation $\dot{\prec}$ is therefore intransitive. On the other hand, the measurement equivalence relation $\doteq$ *is* transitive via the bijection mapping it establishes; e.g., $\texttt{day} \doteq \texttt{dow}$ (day of week).

Proposition 10. *Let u_1/u_2 be two temporal units and $\mathcal{P}_{u_1}$ / $\mathcal{P}_{u_2}$ be the respective set of the non-null periods they map to; i.e., $\mathcal{P}_{u_1} = \{T^p_{u_1}(v_i) \neq \emptyset \mid p \in \mathbb{N}, v_i \in V_{u_1}\}$ and $\mathcal{P}_{u_2} = \{T^q_{u_2} \neq \emptyset \mid q \in \mathbb{N}, v_j \in V_{u_2}\}$. Then the measurement equivalence relation $u_1 \doteq u_2$ establishes a bijection mapping between $\mathcal{P}_{u_1}$ and $\mathcal{P}_{u_2}$.*

With a partial ordering of temporal units specified via the measurement relation, we can then define *granularity* of a coordinate as the finest resolution the coordinate can distinguish on the timeline. For example, the granularity of $c = \{\texttt{2006}_{\texttt{year}}, \texttt{jan}\}$ should be `month` since its time mapping $T(c) = T_{\texttt{year}}(\texttt{2006}) \cap T_{\texttt{month}}(\texttt{jan})$ is resolved to a period $T^q_{\texttt{month}}(\texttt{jan})$ ($q \in \mathbb{N}$) because $\texttt{month} \dot{\leq} \texttt{year}$. It is defined below.

Definition 11. The *granularity* of a coordinate c, written as $\mathrm{g}(c)$, is a set of minimal units of the scope of c under the measurement relation; i.e., $\mathrm{g}(c) = \min(\mathcal{S}(c))$.

For example, using the measurement relation illustrated earlier, we have

$$\mathrm{g}(\{\texttt{2006}_{\texttt{year}}, \texttt{feb}\}) = \{\texttt{month}\}$$
$$\mathrm{g}(\{\texttt{2006}_{\texttt{year}}, 3_{\texttt{qoy}}, \texttt{fall}_{\texttt{sem}}\}) = \{\texttt{qoy}, \texttt{sem}\}$$

Note that in the second example the granularity is a set containing more than one unit. We could have defined granularity to be the greatest lower bound unit of a scope under the measurement relation, so that the granularity of the second example would have been a single unit `month` (since $\texttt{month} \dot{\prec} \texttt{qoy}$ and $\texttt{month} \dot{\prec} \texttt{sem}$). We chose the current definition because in comparison it preserves more information that is directly available from a coordinate.

Periodicity Relation

In addition to the "downward" relation that is measurement (i.e., we can think of measurement as a relation linking the most coarse unit "down" to the finest unit), there is a "upward" relation in force among temporal units. Consider again the expression "*the next January*". We can derive its paraphrase "*the January of the next year*" if we can recognize `month` to be a unit *periodic* in the unit `year`; i.e., every year has one January and one January only. In comparison, the expression "*the January of the next quarter*" is *not* a valid paraphrase because not every quarter of a year has a January, i.e., unit `month` is not periodic in unit `qoy`. Contrasting to the measurement relation $\texttt{month} \dot{\prec} \texttt{year}$ and $\texttt{month} \dot{\prec} \texttt{qoy}$ we can conclude that *periodicity* is indeed a distinct relation from the measurement relation. We now formalize the notion as follows.

Definition 12. Let u_1 and u_2 be two temporal units with values V_{u_1} and V_{u_2}, respectively. We say u_2 *is periodic in* u_1, written as $u_2 \rightarrowtail u_1$, if (a) $u_2 \dot{\prec} u_1$; and (b) for every $q \in \mathbb{N}$ there exists a *unique* $p \in \mathbb{N}$ and $v_i \in V_{u_1}$ such that

$$\bigcup_{v_j \in V_{u_2}} T^q_{u_2}(v_j) \subseteq T^p_{u_1}(v_i).$$

For examples we have $\texttt{day} \rightarrowtail \texttt{month}$ and $\texttt{month} \rightarrowtail \texttt{year}$, but we have $\texttt{month} \not\rightarrowtail \texttt{qoy}$ because not every quarter has every possible month, a violation to the requirement (b). We also have $\texttt{day} \not\rightarrowtail \texttt{year}$, because $\texttt{day} \not\dot{\prec} \texttt{year}$, a violation to the requirement (a).

Since $u_1 \rightarrowtail u_2$ means every period of u_2 contains one (and only one) period of every possible value of u_1, if we are given a particular period of u_2 and a value of u_1, we should be able to identify a unique period of u_1 that corresponds to the value. For example, given the period $T^1_{\texttt{year}}(\texttt{2006})$ and a value `jan`, we should be able to find a unique period $T^q_{\texttt{month}}(\texttt{jan})$ ($q \in \mathbb{N}$) on the timeline that corresponds to the overall coordinate $\{\texttt{2006}_{\texttt{year}}\texttt{jan}\}$. This is stated as a corollary below.

Corollary 13. *Let u_1 and u_2 be two temporal units with values V_{u_1} and V_{u_2}, respectively. If $u_1 \rightarrowtail u_2$ then given a period $T^p_{u_2}(v_i)$ and $v_j \in V_{u_1}$ where $p \in \mathbb{N}$ and $v_i \in V_{u_2}$, there exists a* unique *$q \in \mathbb{N}$ such that $T^q_{u_1}(v_j) \subseteq T^p_{u_2}(v_i)$.*

Consequently if a consistent coordinate assigns a unique value for every unit along a path $u_1 \rightarrowtail u_2 \rightarrowtail \ldots \rightarrowtail u_n$ with u_n being a maximal unit, then its time mapping can be uniquely identified. This is the intuition behind the concept of *anchoring*, to be discussed below.

Anchoring Coordinates

One obvious result from Definition 4 and 5 is that not every coordinate can be mapped to a unique interval on the timeline. For example, a mere $\{\texttt{feb}, 1_{\texttt{day}}\}$ can only map to a set of disjoint intervals, each one denoting a February 1 in a particular year. Since these ambiguous coordinates cannot participate in certain operations such as coordinate comparison, we need to design a procedure to test if a coordinate is one of them. We will first define the concept of an anchored coordinate with respect to our semantic model.

Definition 14. Assume that $\text{glb}(\mathcal{U}_i)$, the greatest lower bound of the set of units $\mathcal{U}_i$ under the measurement relation, always exists, and let c be a consistent coordinate and $u = \text{glb}(\text{g}(c))$. We say c is *anchored* if $T(c) = \{t\}$ where t is a non-null concatenation of *consecutive* value periods of u, namely,

$$t = \bigcup_{q \in Q \subset \mathbb{N}} \bigcup_{v_i \in V_q \subseteq V_u} T_u^q(v_i)$$

where $V_u = \{v_1, \ldots, v_n\}$, $1 \leq i \leq n$, and the set $\{(q-1) \cdot n + i\}$ is an interval of $\mathbb{N}$.

For examples, the coordinate $c_1 = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{feb}\}$ is anchored because $T(c_1) = T^q_{\mathtt{month}}(\mathtt{feb})$ ($|Q| = |V_q| = 1$), and $c_2 = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{3}_{\mathtt{qoy}}, \mathtt{fall}_{\mathtt{sem}}\}$ is anchored because $T(c_2) = T^q_{\mathtt{month}}(\mathtt{aug}) \cup T^q_{\mathtt{month}}(\mathtt{sep})$ ($|Q| = 1$ and $\{(q-1) \cdot 12 + 8, (q-1) \cdot 12 + 9\}$ is an interval of $\mathbb{N}$).

To devise a syntactic procedure for testing the anchoring status of a coordinate, we first observe from Corollary 13 that if a coordinate is an instantiation of a chain of periodic units led by a maximal unit, we can always find a unique period of the sequence' finest unit such that the period is the time mapping of the coordinate, i.e., the coordinate *is* anchored. We therefore call this sequence an *anchor path*.

Corollary 15. *We call a sequence of temporal units* $p = \langle u_n, \ldots, u_1 \rangle$ *an* anchor path *of unit* u_1 *if* $u_1 \rightarrowtail u_2 \rightarrowtail \ldots \rightarrowtail u_n$ *and* u_n *is a maximal unit. A consistent coordinate* c *is said to be* anchored on p *or* anchored at u_1 *if* $p \subseteq \mathcal{S}(c)$*; i.e.,* $\pi_{u_i}(c)$ *is defined for* $1 \leq i \leq n$*.*

For example, $p = \langle \mathtt{year}, \mathtt{month} \rangle$ is an anchor path, but $\langle \mathtt{year}, \mathtt{day} \rangle$ is not. The coordinate $\{\mathtt{2006}_{\mathtt{year}}, \mathtt{feb}\}$ is therefore anchored on p. The decision method for determining the anchoring status of a coordinate is described below.

Proposition 16. *A consistent coordinate* c *is anchored iff for every* $u_i \in \text{g}(c)$*,* c *is anchored on an anchor path* p_i *of* u_i*. Additionally if* $\text{g}(c) = \{u\}$ *then* $T(c) = \{T^q_u(\pi_u(c))\}$*.*

Constructing a Calendar

With calendar CSPs and the two inter-unit relations – measurement and periodicity – in our toolbox, we can now construct a cohesive structure for modeling calendars. But instead of proposing a monolithic structure encompassing all imaginable temporal units and constraints, we opt for a more modular approach. A calendar structure is assembled from a set of *calendar components*: each component has its own minimal and maximal temporal unit, and every component is related to another by an *alignment constraint*. This modular approach has the following advantages:

1. This organization fits with our intuition about calendars. Consider unit `week` in the Gregorian calendar: although $\mathtt{day} \mathrel{\dot{\leq}} \mathtt{week}$ is true, we cannot assert $\mathtt{day} \rightarrowtail \mathtt{week}$ since the "days" periodic in `week` have a different "cycle" (7 instead of 31). `week` also has to be a maximal unit since no other temporal unit in daily use can be measured by

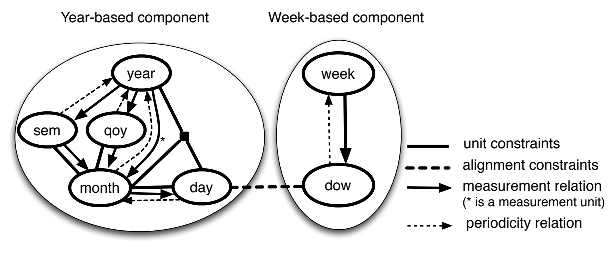

Figure 3: Two aligned calendar components

it. By introducing a separate component for `week` and its company, we are essentially creating a different counting system over the same timeline.
2. With different counting systems delegated to different calendar components, we can "shut down" certain components for efficiency during constraint propagation if the components are deemed irrelevant.
3. New counting systems can be introduced without much disruption with the existing components.

We will now introduce calendar components and describe how to align them to form a full calendar.

Calendar Components

A calendar component is essentially a well-defined calendar CSP souped up with the measurement relation and the periodicity relation. It must also observe several requirements to ensure the unique existence of lower bound and upper bound units and the anchorability of each unit.

Definition 17. A *calendar component* is a triple $(P, \dot{\leq}, \rightarrowtail)$ where $P = (\mathcal{U}, \mathcal{D}, \mathcal{R})$ is a well-defined calendar CSP, $\dot{\leq}$ is the measurement relation on units $\mathcal{U}$ and $\rightarrowtail$ is the periodicity relation on $\mathcal{U}$. All calendar components must meet the following requirements:

1. (*lattice*) the partially ordered set $(\mathcal{U}, \dot{\leq})$ forms a lattice; i.e., for any two units u_1 and u_2 there exists a unique least upper bound u_{sup} such that $u_1 \mathrel{\dot{\leq}} u_{\text{sup}}$ and $u_2 \mathrel{\dot{\leq}} u_{\text{sup}}$, and a unique greatest lower bound u_{sub} such that $u_{\text{sub}} \mathrel{\dot{\leq}} u_1$ and $u_{\text{sub}} \mathrel{\dot{\leq}} u_2$; the top/bottom unit of the component is denoted as $u_{\max}/u_{\min}$, respectively;
2. (*anchorability*) for every unit $u \in \mathcal{U}$ there must exist an anchor path $p = \langle u_{\max}, \ldots, u \rangle$.

Fig. 3 shows two calendar components. Note that some units can be measured by multiple units (e.g., `year`), and we mark one of them (by an asterisk) as the representative measurement unit.

An important consequence of Definition 17 is that a calendar component actually establishes a bijection mapping between its solutions and the set of time mappings at its finest resolution.

Theorem 18. *Consider the calendar component* $(P, \dot{\leq}, \rightarrowtail)$ *where* $P = (\mathcal{U}, \mathcal{D}, \mathcal{R})$ *is a calendar CSP. Then* P *establishes a bijection mapping between the set of solutions to* P *and the time mapping* $T_{u_{\min}}$*, namely*

1. *if* c *is a solution to* P *and* $v = \pi_{u_{\min}}(c)$*, then* $T(c)$ *contains a unique non-null period in* $T_{u_{\min}}(v)$*; i.e.,* $T(c) = \{T^q_{u_{\min}}(v)\}$ *with a unique* $q \in \mathbb{N}$*;*

2. *if $T^p_{u_{\min}}(v)$ is a non-null period where $p \in \mathbb{N}$ and $v \in V_{u_{\min}}$, then there exists a unique solution c such that $\pi_{u_{\min}}(c) = v$ and $T(c) = \{T^p_{u_{\min}}(v)\}$.*

Calendar as Aligned Components

A calendar consists of a set of calendar components, with each one of them representing a different counting system over the same span of time. The components are related to one another via *alignment* constraints, which are used essentially to translate a coordinate of one component to its counterpart in another component. For example, if we have a year-based component and a week-based component as depicted in Fig 3, their alignment must capture the fact that January 6, 2006 *is* the Friday of the 1046160-th week (counting from January 1, 1 AD). We define alignments as follows.

Definition 19. Two calendar components with calendar CSP $P_1 = (\mathcal{U}_1, \mathcal{D}_1, \mathcal{R}_1)$ and $P_2 = (\mathcal{U}_2, \mathcal{D}_2, \mathcal{R}_2)$ can be aligned at unit $u_i \in \mathcal{U}_1$ and $u_j \in \mathcal{U}_2$ using an *alignment constraint* $\mathcal{A}_{u_i,u_j}$ provided the following requirements are met:

1. (*scope*) the scope of $\mathcal{A}_{u_i,u_j}$ is $p_i \cup p_j$, where p_i/p_j are anchor paths of units u_i/u_j, respectively;
2. (*measurement equivalence*) u_i and u_j are measurement equivalent; i.e., $u_i \doteq u_j$.

The unit u_i and u_j are called the *alignment units* of their respective calendar component.

The second requirement not only sets up the desired bijection mapping between T_{u_i} and T_{u_j} (Proposition 10), it also bridges the measurement relation and the periodicity relation across the two calendar components.

Corollary 20. *Let $K_1 = (P_1, \dot{\leq}_1, \rightarrowtail_1)$ and $K_2 = (P_2, \dot{\leq}_2, \rightarrowtail_2)$ be two calendar components aligned at units $u_i \in \mathcal{U}_1$ and $u_j \in \mathcal{U}_2$. The overall measurement relation, $\dot{\leq}$, and the overall periodicity relation, $\rightarrowtail$, are the union of the individual relations ($\dot{\leq}_1 \cup \dot{\leq}_2$ and $\rightarrowtail_1 \cup \rightarrowtail_2$ respectively) plus the following:*

1. *if $u_i \dot{\leq}_1 u_k$ then $u_j \dot{\leq} u_k$, and symmetrically if $u_j \dot{\leq}_2 u_l$ then $u_i \dot{\leq} u_l$;*
2. *if $u_k \dot{\leq}_1 u_i$ then $u_k \dot{\leq} u_j$, and symmetrically if $u_l \dot{\leq}_2 u_j$ then $u_l \dot{\leq} u_i$;*
3. *if $u_k \rightarrowtail_1 u_i$ then $u_k \rightarrowtail u_j$, and symmetrically if $u_l \rightarrowtail_2 u_j$ then $u_l \rightarrowtail u_i$.*

where $u_k \in \mathcal{U}_1$ and $u_l \in \mathcal{U}_2$.

For example, if we add a unit `hour` $\dot{\leq}$ `day` in the year-based component shown in Fig. 3, it is obvious that we should also have `hour` $\dot{\leq}$ `dow`. In addition, `day` $\dot{\leq}$ `month` should imply `day` $\dot{\leq}$ `week`. On the other hand, `hour` $\rightarrowtail$ `day` entails `hour` $\rightarrowtail$ `dow`, but `day` $\rightarrowtail$ `month` does *not* imply `day` $\rightarrowtail$ `week` – because not every `week` has 1_{day} (the first day of a month). This last entailment does not work since Proposition 10 only establishes a bijection mapping on *periods* of the alignment units – it does not change the ways their *values* are cyclic on the timeline.

We are now ready to show how to construct a calendar out of a set of calendar components.

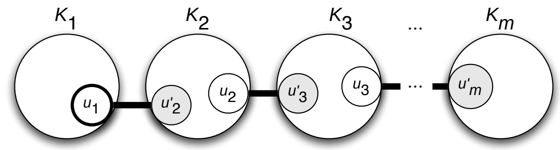

Figure 4: Grounded alignments: u_1 is not a bottom unit, but $u'_2 \ldots u'_m$ must be bottom units

Definition 21. A *calendar* K is a triple $(P, \dot{\leq}, \rightarrowtail)$ where $P = (\mathcal{U}, \mathcal{D}, \mathcal{R})$ is a calendar CSP, $\dot{\leq}$ is the measurement relation on units $\mathcal{U}$ and $\rightarrowtail$ is the periodicity relation on $\mathcal{U}$. K is composed of a set of aligned calendar components $K_i = (P_i, \dot{\leq}_i, \rightarrowtail_i)$ where $P_i = (\mathcal{U}_i, \mathcal{D}_i, \mathcal{R}_i)$ and $i = 1 \ldots n$, and the following requirements must be observed:

1. (*tree*) if we view K as a undirected graph where nodes are the calendar components and edges are the alignments, then K must be a simple connected graph with no cycle;
2. (*grounded alignments*) if unit $u_1 \in \mathcal{U}_1$ in an alignment $\mathcal{A}_{u_1,u'_2}$ is not the bottom unit of the component K_1, then on the longest path of the aligned components $\langle K_1, \ldots, K_m \rangle$ with the alignments $\mathcal{A}_{u_1,u'_2}, \mathcal{A}_{u_2,u'_3}, \ldots, \mathcal{A}_{u_{m-1},u'_m}$, unit $u'_2, \ldots, u'_m$ must be the bottom unit of the component $K_2, \ldots, K_m$, respectively (Fig. 4).

Finally let $\mathcal{A}$ be the set of all alignment constraints in K, the construction of K is specified as follows:

$$
\begin{aligned}
&\mathcal{U} = \mathcal{U}_1 \cup \ldots \cup \mathcal{U}_n \\
&\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_n \\
&\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_n \cup \mathcal{A} \\
&\dot{\leq} \;=\; \dot{\leq}_1 \cup \ldots \cup \dot{\leq}_n \cup \\
&\qquad (\cup_{\mathcal{A}_{u_1,u_2} \in \mathcal{A}} \{(u_2, u'_1) \mid u_1 \dot{\prec}_i u'_1\} \cup \{(u_1, u'_2) \mid u_2 \dot{\prec}_j u'_2\} \cup \\
&\qquad \{(u'_1, u_2) \mid u'_1 \dot{\prec}_i u_1\} \cup \{(u'_2, u_1) \mid u'_2 \dot{\prec}_j u_2\}) \\
&\rightarrowtail \;=\; \rightarrowtail_1 \cup \ldots \cup \rightarrowtail_n \cup \\
&\qquad (\cup_{\mathcal{A}_{u_1,u_2} \in \mathcal{A}} \{(u'_1, u_2) \mid u'_1 \rightarrowtail_i u_1\} \cup \{(u'_2, u_1) \mid u'_2 \rightarrowtail_j u_2\})
\end{aligned}
$$

Note that the constructions of $\dot{\leq}$ and $\rightarrowtail$ above basically follow Corollary 20, except that we are patching up $\dot{\leq}$ using the individual $\dot{\prec}_i$ relations for efficiency reason (because the measurement relation is transitive). The first requirement above (tree structure) is needed to simplify the model and to support some of the proofs shown in the appendix. The second requirement (grounded alignments) is imposed to ensure the existence of a unique greatest lower bound (up to measurement equivalence) given any pair of units, as asserted by the following theorem.

Theorem 22. *A calendar K constructed in Definition 21 has the following properties:*

1. *(semilattice) for any two units in $\mathcal{U}$ there exists a unique greatest lower bound up to measurement equivalence;*
2. *(anchorability) for every unit u in $\mathcal{U}$ there exists an anchor path of u;*
3. *(well-defined calendar CSP) The calendar CSP of K, P, is well-defined;*

4. *(modularity) if c is a consistent coordinate whose scope is within a single calendar component $K_i = (P_i, \dot{\leq}_i, \rightarrowtail_i)$, then for every solution $c_i \in \mathcal{E}(c)$ with respect to P_i, $\mathcal{E}(c_i) \neq \emptyset$ with respect to P.*

A corollary of Theorem 22 is that Theorem 18 can now be carried over to a calendar as well.

Corollary 23. *Let P be the calendar CSP of a calendar, then Theorem 18 applies to P as well.*

A final observation of Theorem 22 is that the modularity property enables an opportunity to enhance efficiency when running the solution methods of a calendar CSP: given a coordinate c that instantiates only units from a single calendar component, we can ignore all of the other components when iterating through $\mathcal{E}(c)$. For example, if $c = \{\mathtt{feb}, 29_{\mathtt{day}}\}$, then propagating any constraint related to the week-based component will not affect $\mathcal{E}(c)$ with respect to only the year-based component; e.g., $\{2006_{\mathtt{year}}, \mathtt{feb}, 29_{\mathtt{day}}\}$ should be consistent whether we propagate the alignment constraint $\mathcal{A}_{\mathtt{day},\mathtt{dow}}$ or not. This is easy to implement by ignoring the unnecessary alignment constraints when performing constraint propagation.

Time Mappings and Extensions

We have introduced earlier the semantic model of a calendar CSP, but deferred the discussion of a semantic interpretation for coordinate extensions and extension projections (Definition 1 and 3). We are now ready to relate the two concepts – the time mappings and the extension/extension projection operations – in the form of the two propositions below. The result will also be helpful to prove the correctness of our method for comparing anchored coordinates (see Theorem 30 and its proof in the appendix).

The first proposition states that the time mapping of a coordinate is the union of the time mapping of every solution in its extension.

Proposition 24. *Let c be a consistent coordinate with respect to a calendar defined in Definition 21, then $T(c) = \bigcup_{c' \in \mathcal{E}(c)} T(c')$.*

For example, we have

$$T(\{2006_{\mathtt{year}}, \mathtt{jan}\}) = T(\{2006_{\mathtt{year}}, \mathtt{jan}, 1_{\mathtt{qoy}}, \mathtt{spring}, 1_{\mathtt{day}}\}) \cup \ldots \cup T(\{2006_{\mathtt{year}}, \mathtt{jan}, 1_{\mathtt{qoy}}, \mathtt{spring}, 31_{\mathtt{day}}\}).$$

A similar result holds for extension projections: the time mapping of a coordinate is the union of the time mapping of every coordinate in its extension projection, as long as the projection is done over an anchor path of a unit "finer" than the granularity of the coordinate.

Proposition 25. *Let c be a consistent coordinate with respect to a calendar defined in Definition 21, and p be an anchor path of unit $u \dot{\leq} \mathrm{glb}(\mathrm{g}(c))$, then $T(c) = \bigcup_{c' \in \mathcal{E}_p(c)} T(c')$.*

For example, if $c = \{2006_{\mathtt{year}}, \mathtt{spring}\}$ and $p_1 = \langle \mathtt{year}, \mathtt{month} \rangle$, we have

$$T(c) = \cup_{c' \in \mathcal{E}_{p_1}(c)} T(c') = T(\{2006_{\mathtt{year}}, \mathtt{jan}\}) \cup \ldots \cup T(\{2006_{\mathtt{year}}, \mathtt{may}\}).$$

But for $p_2 = \langle \mathtt{year}, \mathtt{qoy} \rangle$, Proposition 25 no longer holds since $\mathtt{qoy} \not\dot{\leq} \mathtt{sem}$:

$$\begin{aligned} T(c) &\subset \cup_{c' \in \mathcal{E}_{p_2}(c)} T(c') \\ &= T(\{2006_{\mathtt{year}}, 1_{\mathtt{qoy}}\}) \cup T(\{2006_{\mathtt{year}}, 2_{\mathtt{qoy}}\}) \\ &= T(\{2006_{\mathtt{year}}, \mathtt{jan}\}) \cup \ldots \cup T(\{2006_{\mathtt{year}}, \mathtt{jun}\}). \end{aligned}$$

Calendar Services

A calendar system must provide useful services to the rest of a system. In this section we describe how some of these services can be implemented under our constraint-based model. Due to space restriction we will not describe services such as searching for a compatible coordinate and performing calendric arithmetic, other than to simply mention that both of them can be implemented using the constraint distribution algorithm mentioned earlier.

Coordinate Comparison

One of the basic services that a calendar model needs to provide is comparing the chronological ordering of two coordinates, namely deciding which of the two is *earlier*. From the semantic point of view, it is to compare their time mappings on the timeline, as defined below.

Definition 26. Let c_1 and c_2 be two consistent coordinates. We say $c_1 < c_2$ if $T(c_1) < T(c_2)$.

As before we would like to find a corresponding "syntactic procedure" that obviates any direct manipulation of time mappings. First we note that such a comparison does not make sense if the two participating coordinates are not anchored. To facilitate our discussion, we define a useful concept called *anchor sets* below.

Definition 27. Let $p = \langle u_n, \ldots, u_1 \rangle$ be an anchor path, then the *anchor set* of p is defined as the set of *consistent* coordinates of scope p, namely $\mathrm{AnchorSet}_p = \{c \mid \mathcal{E}(c) \neq \emptyset \text{ and } \mathcal{S}(c) = p\}$.

Obviously by Proposition 16 every coordinate in an anchor set is anchored on the same anchor path. If both of the coordinates in a comparison belong to the same anchor set, then an intuitive way to compare the two is to compare their value assignments *lexicographically* from the maximal unit downward along the anchor path. For example, we can conclude that $\{2006_{\mathtt{year}}, \mathtt{jan}\}$ is before $\{2006_{\mathtt{year}}, \mathtt{feb}\}$ because we have $2006 = 2006$ and $\mathtt{jan}$ is before $\mathtt{feb}$, in that order. This procedure is formalized as follows.

Definition 28. Let $p = \langle u_n, \ldots, u_1 \rangle$ be an anchor path and $c_1, c_2 \in \mathrm{AnchorSet}_p$. We say $c_1 <_p c_2$ if there exists $1 \leq k \leq n$ such that $\pi_{u_i}(c_1) = \pi_{u_i}(c_2)$ for all $i > k$ and $\pi_{u_k}(c_1) < \pi_{u_k}(c_2)$; $c_1 =_p c_2$ if $\pi_p(c_1) = \pi_p(c_2)$.

It can be shown that the lexicographic comparison indeed establishes the ordering of the two coordinates on the timeline, as asserted in the proposition below.

Proposition 29. *Let $p = \langle u_n, \ldots, u_1 \rangle$ be an anchor path and $c_1, c_2 \in \mathit{AnchorSet}_p$, then $c_1 <_p c_2$ iff $T(c_1) < T(c_2)$ and $c_1 =_p c_2$ iff $T(c_1) = T(c_2)$.*

Obviously the lexicographic test described in Definition 28 is useless when the two coordinates in a comparison are not anchored on the same anchor path (e.g., comparing $\{\mathtt{2006}_{\mathtt{year}}, \mathtt{1}_{\mathtt{qoy}}\}$ with $\{\mathtt{2006}_{\mathtt{year}}, \mathtt{fall}_{\mathtt{sem}}\}$). For these general cases we need to compute a common anchor path ending at the greatest lower bound of the two granularities, and to compare the projections of the two coordinates on this anchor path. This is formalized in the following theorem.

Theorem 30. *Let c_1 and c_2 be two consistent anchored coordinates and p be an anchor path such that $\min(p) = \mathrm{glb}(\mathrm{g}(c_1) \cup \mathrm{g}(c_2))$, then $c_1 < c_2$ iff $c'_1 <_p c'_2$ for every $c'_1 \in \mathcal{E}_p(c_1)$ and $c'_2 \in \mathcal{E}_p(c_2)$, and $c_1 = c_2$ iff $=_p$ establishes a bijective mapping between $\mathcal{E}_p(c_1)$ and $\mathcal{E}_p(c_2)$.*

For example, to compare $c_1 = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{1}_{\mathtt{qoy}}\}$ with $c_2 = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{fall}_{\mathtt{sem}}\}$ we first compute $\mathrm{glb}(\mathtt{qoy}, \mathtt{sem}) = \mathtt{month}$ and then the anchor path $p = \langle \mathtt{year}, \mathtt{month} \rangle$ (see Fig. 3). Using a constraint distribution algorithm we can then compute the latest coordinate in $\mathcal{E}_p(c_1)$ to be $c'_1 = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{mar}\}$ and the earliest coordinate in $\mathcal{E}_p(c_2)$ to be $c'_2 = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{aug}\}$. Since $c'_1 <_p c'_2$ we conclude $c_1 < c_2$.

Granularity Conversion

Intuitively speaking, converting granularity of a coordinate is to look at the same coordinate at a new granularity; for example, when looking at February 2006 on a timeline at granularity day, we see that February 1-28, 2006 are the dates included in the original one. On the other hand, if we look at February 1, 2006 at granularity month, we will see that the original date becomes February 2006 at that granularity. It turns out that this intuition is already embodied in Definition 3: for example in converting $c = \{\mathtt{2006}_{\mathtt{year}}, \mathtt{feb}\}$ from granularity $\{\mathtt{month}\}$ to $\{\mathtt{day}\}$, we are essentially computing $\mathcal{E}_\Theta(c)$ where $\Theta = \{\mathtt{year}, \mathtt{month}, \mathtt{day}\}$. Obviously for any $c' \in \mathcal{E}_\Theta(c)$ we are guaranteed to have the right granularity $\mathrm{g}(c') = \min(\mathcal{S}(c')) = \min(\Theta) = \{\mathtt{day}\}$. This is formalized as follows.

Definition 31. Given a coordinate c and a target granularity γ, the *granularity conversion function* $\mathrm{gconv}(c, \gamma)$ returns $\mathcal{E}_\Theta(c)$ where Θ is computed as:

$$\Theta = \cup_{K_i}(\Phi_i \cup \Psi_i \cup \Gamma_i)$$

$$\Phi_i = \{u' \mid u \dot{\preceq} u', u \in \gamma_i, u' \in \mathcal{S}_i(c)\} \tag{1}$$

$$\Psi_i = \{u', u'', u \mid u' \rightarrowtail \ldots \rightarrowtail u'' \rightarrowtail \ldots \rightarrowtail u, u' \in \gamma'_i, u \in \mathrm{g}_i(c)\} \tag{2}$$

$$\Gamma_i = \begin{cases} \{u', u'', u_{\max} \mid u' \rightarrowtail \ldots \rightarrowtail u'' \rightarrowtail \ldots \rightarrowtail u_{\max}, \\ \quad u' \in \gamma_i \setminus \gamma'_i\} & \text{if } c \text{ is anchored at } \mathrm{g}(c); \\ \gamma_i \setminus \gamma'_i & \text{otherwise.} \end{cases} \tag{3}$$

where K_i is a calendar component, $\mathcal{U}_i$ is the set of units in the component K_i, $\mathcal{S}_i(c) = \mathcal{S}(c) \cap \mathcal{U}_i$, $\gamma_i = \gamma \cap \mathcal{U}_i$, $\gamma'_i = \{u' \mid u' \rightarrowtail \ldots \rightarrowtail u, u' \in \gamma_i, u \in \mathrm{g}_i(c)\}$ and $\mathrm{g}_i(c) = \mathrm{g}(c) \cap \mathcal{U}_i$.

The granularity conversion function defined above essentially returns the extension projection of the input coordinate on a new scope Θ. Fig. 5 illustrates how the new scope is computed. For every calendar component K_i, (1) *removes*

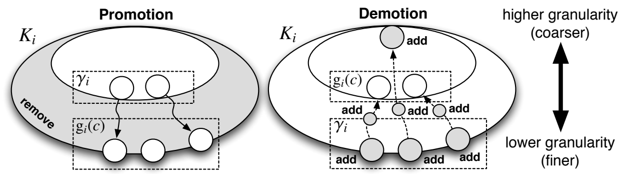

Figure 5: Granularity conversion of a coordinate c on a single calendar component: component-wise the original granularity is $\mathrm{g}_i(c)$ and the target granularity is γ_i. The solid arrows represent the measurement relation, and the dashed arrows represent the periodicity relation. This illustration shows the case where c is anchored at $\mathrm{g}(c)$.

any unit in the original scope ($\mathcal{S}_i(c)$, the scope of c in K_i) that is not measured by any unit in the target granularity (γ_i, the target granularity specific to K_i). (2) then *adds* to the new scope any unit along the anchor path of $u \in \gamma_i$ up to a unit in $\mathrm{g}_i(c)$ (the original granularity specific to K_i). Finally (3) adds those units in the target granularity that do not have anchor paths passing through any unit in the original granularity: if c is originally anchored, the equation adds entire anchor paths to maintain the anchoring status of the coordinate, otherwise only the units in γ'_i are added. In summary, (1) promotes granularity by shrinking the scope, while (2) and (3) demote granularity by enlarging the scope.

The correctness of Definition 31 is stated below, which can be easily verified by observing from the definition that every unit in Θ must be measured by a unit in γ.

Corollary 32. *Let c be a coordinate and γ be a set of minimal units, then for every $c' \in \mathrm{gconv}(c, \gamma)$, $\mathrm{g}(c') = \gamma$.*

For examples, we have the following conversions

$$\mathrm{gconv}(\{\mathtt{2006}_{\mathtt{year}}, \mathtt{feb}\}, \{\mathtt{year}\}) = \{\{\mathtt{2006}_{\mathtt{year}}\}\} \tag{4}$$

$$\mathrm{gconv}(\{\mathtt{2006}_{\mathtt{year}}, \mathtt{1}_{\mathtt{qoy}}\}, \{\mathtt{sem}\}) = \{\{\mathtt{2006}_{\mathtt{year}}, \mathtt{spring}\}\} \tag{5}$$

$$\mathrm{gconv}(\{\mathtt{2006}_{\mathtt{year}}\}, \{\mathtt{day}\}) = \{\{\mathtt{2006}_{\mathtt{year}}, \mathtt{jan}, \mathtt{1}_{\mathtt{day}}\}, \ldots, \{\mathtt{2006}_{\mathtt{year}}, \mathtt{dec}, \mathtt{31}_{\mathtt{day}}\}\} \tag{6}$$

In the examples above, (4) shows granularity promotion (month is removed from the scope) while (5) shows both promotion (qoy is removed from the scope) and demotion (sem is added to the scope). When a demotion involves a target unit that is "far" from the units in the original granularity, more units along the relevant anchor paths are added to the scope to make the converted coordinates more "granularity-smoothed", e.g., month is added to the scope in (6).

Summary

In this paper we have presented a constraint-based model of calendars, where temporal units are viewed as variables with fully-ordered domains, and constraints are given to relate the units. The resulting calendar constraint satisfaction problem is augmented with the measurement relation and the periodicity relation to form calendar components, which can then be aligned via constraints to construct a full calendar. We also introduced a semantic model to justify the imposed design requirements over the calendar system, and described selected calendar services such as time comparison and granularity conversion.

A complete system using the proposed calendar system has been implemented and successfully deployed in natural language applications such as “normalizing” temporal expressions found in news articles and web blogs (Florian et al. 2007). One possible future work is to explore the connections between our approach and the algebraic representation proposed in (Peng Ning, Wang, and Jajodia 2002), e.g., how to infer constraints among temporal units using an algebraic representation.

References

Bettini, C., and Sibi, R. D. 2000. Symbolic representation of user-defined time granularities. *Annals of Mathematics and Artificial Intelligence*.

Combi, C.; Franceschet, M.; and Peron, A. 2004. Representing and reasoning about temporal granularities. *Journal of Logic and Computation*.

Dechter, R. 2003. *Constraint Processing*. Morgan Kaufmann.

Florian, R.; Han, B.; Luo, X.; Kambhatla, N.; and Zitouni, I. 2007. IBM ACE’07 System Description. In *Proceedings of NIST 2007 Automatic Content Extraction Evaluation*.

Han, B.; Gates, D.; and Levin, L. 2006. From Language to Time: A Temporal Expression Anchorer. In *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME 2006)*.

Mackworth, A. K. 1977. Consistency in networks of relations. *Artificial Intelligence* 8:99–118.

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

Peng Ning, X.; Wang, S.; and Jajodia, S. 2002. An algebraic representation of calendars. *Annals of Mathematics and Artificial Intelligence* 36(1-2):5–38.

Wijsen, J. 2000. A string-based model for infinite granularities. In *The AAAI-2000 Workshop on Spatial and Temporal Granularity*, 9–16. AAAI Press.

Appendix: Selected Proofs

Proof of Theorem 18.

1. Since c is a solution, by Corollary 15 c must be anchored at $u_{\min}$, and by Proposition 16 we must have $T(c) = \{T^{q}_{u_{\min}}(v)\}$ since $g(c) = \{u_{\min}\}$.
2. By the second requirement of a well-defined calendar CSP (Definition 7) we know there exists a consistent coordinate c for $T^{p}_{u_{\min}}(v)$. Assume there are two different solutions $c_1, c_2 \in \mathcal{E}(c)$, then according to the first part of this proof we have $T(c_1) = \{T^{q_1}_{u_{\min}}(\pi_{u_{\min}}(c_1))\}$ and $T(c_2) = \{T^{q_2}_{u_{\min}}(\pi_{u_{\min}}(c_2))\}$ $(q_1, q_2 \in \mathbb{N})$. Since $c \subseteq c_1$, from Corollary 6 we must have $T^{q_1}_{u_{\min}}(\pi_{u_{\min}}(c_1)) \subseteq T^{p}_{u_{\min}}(v)$, which can only be true if $T^{q_1}_{u_{\min}}(\pi_{u_{\min}}(c_1)) = T^{p}_{u_{\min}}(v)$, $q_1 = p$ and $\pi_{u_{\min}}(c_1) = v$. By similar reasoning we have $T^{q_1}_{u_{\min}}(\pi_{u_{\min}}(c_1)) = T^{q_2}_{u_{\min}}(\pi_{u_{\min}}(c_2)) = T^{p}_{u_{\min}}(v)$, a violation to Corollary 8 since $c_1 \neq c_2$. Therefore $c_1 = c_2$ and $\pi_{u_{\min}}(c_1) = \pi_{u_{\min}}(c_2) = v$.

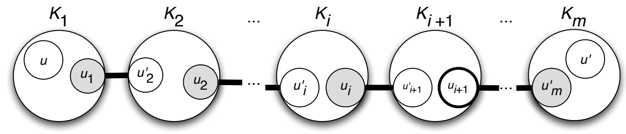

Figure 6: Proof of unique greatest lower bound.

□

Proof of Theorem 22.

1. Given two units u and u', if both of them are in the same calendar component then they must have a unique greatest lower bound (see the lattice requirement in Definition 17). If u and u' belongs to two different components K_1 and K_m, respectively, since all calendar components are aligned and there is no alignment cycle, we can find a unique path $\langle K_1, \dots, K_m \rangle$ connected by the alignments $\mathcal{A}_{u_1,u'_2}, \mathcal{A}_{u_2,u'_3}, \dots, \mathcal{A}_{u_{m-1},u'_m}$. We then have the following cases:
   (a) If u_1 is *not* the bottom unit of component K_1, then according to the grounded alignments requirement in Definition 19, unit $u'_2, \dots, u'_m$ must be the bottom unit of $K_2, \dots, K_m$, and we have $u_1 \doteq u'_2 \dot{\leq} u_2 \doteq \dots u'_m \dot{\leq} u'$. Therefore $\mathrm{glb}(u, u') = \mathrm{glb}(u, u_1)$.
   (b) If u_1 is the bottom unit of component K_1, let $\langle K_1, \dots, K_i \rangle$ be the longest path such that $u_1, \dots, u_i$ are all bottom units of their respective components ($i < m$; see Fig. 6). Then u_{i+1} must not be a bottom unit, and according to the proof in (a) we know for $u'' \in \mathcal{U}_{i+1}$, $\mathrm{glb}(u'', u') = \mathrm{glb}(u'', u_{i+1})$. From (a) we also know $\mathrm{glb}(u, u'') = \mathrm{glb}(u'_{i+1}, u'')$, therefore $\mathrm{glb}(u, u') = \mathrm{glb}(u'_{i+1}, u_{i+1})$.
2. See the anchorability requirement in Definition 17.
3. We need to prove that both of the requirements in Definition 7 are met.
   (a) We need to show that given a solution c to P there exists a unique non-null period t such that $T(c) = \{t\}$. Let us first consider two calendar components K_1 and K_m in the calendar. From the first part of this proof we know K_1 and K_m must be connected by a path of components aligned in the way shown in Fig. 6 ($u_1, u_2, \dots, u_i$ and $u'_{i+2}, u'_{i+3}, \dots, u'_m$ are bottom units; u_{i+1} is *not* a bottom unit). Since c is a solution to P, $c_i = \pi_{\mathcal{U}_i}(c)$ must be a solution to P_i – the calendar CSP of the component K_i. Additionally because P_i is presumed to be a well-defined CSP, by Theorem 18 we have $T(c_i) = \{T^{p_i}_{u_{\min,i}}(\pi_{u_{\min,i}}(c))\}$ with unique $p_i \in \mathbb{N}$ and $u_{\min,i} = \mathrm{glb}(\mathcal{U}_i)$. Based on the topology shown in Fig. 6 and the proposition of measurement equivalence relation (Proposition 10), we then have

$$T^{p_{i+1}}_{u_{\min,i+1}}(\pi_{u_{\min,i+1}}(c)) \subseteq T^{q'_{i+1}}_{u'_{i+1}}(\pi_{u'_{i+1}}(c)) = T^{p_i}_{u_i}(\pi_{u_i}(c)) \subseteq T^{q'_i}_{u'_i}(\pi_{u'_i}(c)) = \dots$$

$$T^{p_{i+1}}_{u_{\min,i+1}}(\pi_{u_{\min,i+1}}(c)) \subseteq T^{q_{i+1}}_{u_{i+1}}(\pi_{u_{i+1}}(c)) = T^{p_{i+2}}_{u'_{i+2}}(\pi_{u'_{i+2}}(c)) \subseteq T^{q_{i+2}}_{u_{i+2}}(\pi_{u_{i+2}}(c)) = \dots$$

where $p_1, \ldots, p_m \in \mathbb{N}$ are period indices for the bottom units, $q'_1, \ldots, q'_{i+1} \in \mathbb{N}$ and $q_{i+1}, \ldots, q_m \in \mathbb{N}$ are the other period indices. Combining the two equations above, we have

$$\cap_{1\leq j\leq m} T(c_j) = \{\cap_{1\leq j\leq m} T^{p_j}_{u_{\min,j}}(\pi_{u_{\min,j}}(c))\} = \{T^{p_{i+1}}_{u_{\min,i+1}}(\pi_{u_{\min,i+1}}(c))\}.$$

We call the component K_{i+1} the "inflection" component among the components $K_1 \ldots K_m$, since its bottom unit is a lower bound of all of the other bottom units, and the unique non-null period $T^{p_{i+1}}_{u_{\min,i+1}}(\pi_{u_{\min,i+1}}(c))$ represents the result of intersecting all of the unique non-null periods mapped by the components $K_1 \ldots K_m$. We can then repeat this process on the pair of the inflection component K_{i+1} and another "unprocessed" component K_k ($k \notin \{1, \ldots m\}$), and find the new inflection component K_l among the set of components aligned between K_{i+1} and K_k – let us call this set of aligned components S. Again $T^{p_l}_{u_{\min,l}}(\pi_{u_{\min,l}}(c))$ ($p_l \in \mathbb{N}$) will represent the result of intersecting the unique non-null period mapped by every component in $S \cup \{K_1 \ldots K_m\}$. Repeating this process over all components in K we can then obtain a unique non-null period t from the final inflection component, and $T(c) = \{t\}$.

(b) From the grounded alignments requirement in Definition 21, we know for any two calendar components K_1 and K_2 aligned with the alignment $\mathcal{A}_{u_1,u_2}$, one of u_1 and u_2 must be a bottom unit. Without loss of generality let u_2 be the bottom unit of K_2.

If we are given a non-null period of unit $u \in \mathcal{U}_1$, say $T^p_u(v_u)$ ($p \in \mathbb{N}$ and $v_u \in V_u$), since P_1 – the calendar CSP of K_1 – is well-defined, according to Definition 7 there must exist a consistent coordinate c (with respect to K_1) such that $T(c) = \{T^p_u(v_u)\}$. Let $c_1 \in \mathcal{E}(c)$ be a solution to K_1, then according to Theorem 18 $T(c_1) = \{T^{p_1}_{\mathrm{glb}(\mathcal{U}_1)}(\pi_{\mathrm{glb}(\mathcal{U}_1)}(c_1))\}$, where the sole period – let us rewrite it as t – is non-null. Therefore we must have $t \subseteq T^p_u(v_u) \cap T^q_{u_1}(v_{u_1})$ ($q \in \mathbb{N}$, $v_u = \pi_u(c_1)$ and $v_{u_1} = \pi_{u_1}(c_1)$). But we already know both t and $T^p_u(v_u)$ are non-null, hence $T^q_{u_1}(v_{u_1})$ is also non-null. Since $u_1 \doteq u_2$, by Proposition 10 we know $T^q_{u_1}(v_{u_1}) = T^{q'}_{u_2}(v_{u_2})$ ($q' \in \mathbb{N}$ and $v_{u_2} \in V_{u_2}$), therefore $T^{q'}_{u_2}(v_{u_2})$ is non-null. Since P_2 is a well-defined calendar CSP and u_2 is a bottom unit, from Theorem 18 we know there exists a solution c_2 such that $T(c_2) = \{T^{q'}_{u_2}(v_{u_2})\}$. Finally we note $c_1 \cup c_2$ must be a solution to the aligned K_1 and K_2, therefore $c \subseteq c_1 \cup c_2$ must be consistent over the aligned K_1 and K_2.

On the other hand, if we are given a non-null period of unit $u' \in \mathcal{U}_2$, say $T^{p'}_{u'}(v_{u'})$ ($p' \in \mathbb{N}$ and $v_{u'} \in V_{u'}$), since P_2 – the calendar CSP of K_2 – is well-defined, according to Definition 7 there must exist a consistent coordinate c' (with respect to K_2) such that $T(c') = \{T^{p'}_{u'}(v_{u'})\}$. Let $c'_2 \in \mathcal{E}(c')$ be a solution to K_2, then according to Theorem 18 $T(c'_2) = \{T^{p'_2}_{\mathrm{glb}(\mathcal{U}_2)}(\pi_{\mathrm{glb}(\mathcal{U}_2)}(c'_2))\}$, where the sole period – let us rewrite it as t' – is non-null. Since $u_1 \doteq u_2$, by Proposition 10 we know $T^{q'}_{u_1}(v'_{u_1}) = t'$ ($q' \in \mathbb{N}$ and $v'_{u_1} \in V_{u_1}$), therefore $T^{q'}_{u_1}(v'_{u_1})$ is non-null. Now because P_1 is a well-defined calendar CSP, from Definition 7 there must exist a consistent coordinate c (with respect to K_1) such that $T(c) = \{T^{q'}_{u_1}(v'_{u_1})\}$. Let $c'_1 \in \mathcal{E}(c)$ be a solution to K_1, then $c'_1 \cup c'_2$ must be a solution to the aligned K_1 and K_2, therefore $c' \subseteq c'_1 \cup c'_2$ must be consistent over the aligned K_1 and K_2.

In the above we have shown both cases when two calendar components are aligned. Similar reasoning can be applied to find a consistent coordinate with respect to all of the aligned components given a non-null period of any unit.

4. Let $\mathcal{A}_{u_i,u_j}$ be an alignment between component K_i and K_j. Since c_i is a solution to P_i, there must exist a non-null period $T^q_{u_i}(\pi_{u_i}(c_i))$ ($q \in \mathbb{N}$) according to Theorem 18. Since $u_i \doteq u_j$ is true, according to Proposition 10 we also have a non-null period $t = T^q_{u_i}(\pi_{u_i}(c_i))$ of unit u_j. Because P_j is also a well-defined CSP, by the second requirement of Definition 7 we know there exists a consistent coordinate c' with respect to P_j such that $T(c') = \{t\}$, hence a solution $c_j \in \mathcal{E}(c')$ exists, and $c_i \cup c_j$ is a solution to the aligned K_i and K_j. By induction we can extend a solution in one component to a solution in another, and the union of these component-wise solutions becomes a solution to P. Therefore c_i can be extended to a solution to P, i.e., $\mathcal{E}(c_i)$ must not be empty.

□

Proof of Theorem 30.

⇒: If $c'_1 <_p c'_2$ for all $c'_1 \in \mathcal{E}_p(c_1)$ and $c'_2 \in \mathcal{E}_p(c_2)$, since $c'_1, c'_2 \in \mathrm{AnchorSet}_p$, from Proposition 29 we know $T(c'_1) < T(c'_2)$. Next since $\min(p) \dot{\leq} \mathrm{glb}(\mathrm{g}(c_1))$ and $\min(p) \dot{\leq} \mathrm{glb}(\mathrm{g}(c_2))$, by Proposition 25 we derive $T(c_1) = \bigcup_{c'_1 \in \mathcal{E}_p(c_1)} T(c'_1) < \bigcup_{c'_2 \in \mathcal{E}_p(c_2)} T(c'_2) = T(c_2)$.

If $=_p$ is a bijective relation between $\mathcal{E}_p(c_1)$ and $\mathcal{E}_p(c_2)$, then from Proposition 29 again we know $T(c'_1) = T(c'_2)$ for every pair of $c'_1 =_p c'_2$. And by Proposition 25 we derive $T(c_1) = \bigcup_{c'_1 \in \mathcal{E}_p(c_1)} T(c'_1) = \bigcup_{c'_2 \in \mathcal{E}_p(c_2)} T(c'_2) = T(c_2)$.

⇐: If $c_1 < c_2$, given $\min(p) \dot{\leq} \mathrm{glb}(\mathrm{g}(c_1))$ and $\min(p) \dot{\leq} \mathrm{glb}(\mathrm{g}(c_2))$, we know from Proposition 25 that $T(c_1) = \bigcup_{c'_1 \in \mathcal{E}_p(c_1)} T(c'_1) < T(c_2) = \bigcup_{c'_2 \in \mathcal{E}_p(c_2)} T(c'_2)$, which implies $T(c'_1) < T(c'_2)$ for all $c'_1 \in \mathcal{E}_p(c_1)$ and $c'_2 \in \mathcal{E}_p(c_2)$. Since $c'_1, c'_2 \in \mathrm{AnchorSet}_p$, by Proposition 29 we know $c'_1 <_p c'_2$.

If $c_1 = c_2$, again from Proposition 25 we know $T(c_1) = \bigcup_{c'_1 \in \mathcal{E}_p(c_1)} T(c'_1) = T(c_2) = \bigcup_{c'_2 \in \mathcal{E}_p(c_2)} T(c'_2)$. Every $T(c'_1)$ must be disjoint since $\mathcal{E}_p(c_1)$ is a set, and according to Proposition 16 $T(c'_1) = \{t\}$ where t is a single value period of unit $\min(p)$. Similarly every $T(c'_2)$ is also disjoint. It is obvious then that for every $T(c'_1)$ there must exist a $T(c'_2)$ such that the two are equal, or by Proposition 29, $c'_1 =_p c'_2$, and the reverse is also true. Therefore $=_p$ is a bijective relation between $\mathcal{E}_p(c_1)$ and $\mathcal{E}_p(c_2)$.

□