Álgebra del Tiempo Continuo

Daniel Rus y Víctor de Buen

17 de agosto de 2005

Resumen

En este documento se dan las principales bases teóricas para la implementación del **álgebra del tiempo continuo** especialmente diseñada para el lenguaje de programación *open-source* TOL (*Time Oriented Language*) cuya página web es TOL-project `http://www.tol-project.org`, y que ha sido desarrollado por el grupo Bayes `http://www.bayesforecast.com`.

El propósito del álgebra del tiempo es dotar al lenguaje de un método potente y sencillo de manejar que permita la manipulación de secuencias temporales complejas utilizadas en el análisis estadístico de series temporales, tanto en tiempo continuo como discreto, especialmente en lo que se respecta al llamado tiempo social que alude a todos aquellos aspectos de tipo social que pueden influir en los fenómenos analizados, como son los festivos, los puentes, las vacaciones, el calendario escolar, fines de semana, estacionalidades anual, semanal, y diaria, usos horarios de invierno y verano, nocturnidad, horario de oficina, etc.; o cualquier otra estructura temporal derivada de los mismos.

Cuando se habla de tiempo continuo en realidad se hace referencia a que las secuencias temporales no tienen porqué ser regulares sino que pueden seguir cualquier sucesión creciente de puntos de la recta real, aunque en la práctica se limitará la precisión temporal a cierta granularidad máxima, que en este caso será el segundo. Es decir, el tiempo manejado no es realmente continuo sino discreto, pero admite secuencias irregulares y con un grado de precisión que se estima más que suficiente por ahora, aunque en todo momento se deja la puerta abierta a precisiones más altas.

Se introducen formalizaciones matemáticas de granularidad temporal, instante de tiempo y conjunto temporal, así como de las operaciones algebraicas que es posible definir sobre ellos.

Índice

1. Granularidad Temporal

La granularidad temporal hace referencia a las unidades básicas o granos de medición del tiempo como el segundo, el minuto, la hora, etc. A continuación se describe el concepto formal de granularidad en la recta real y su aplicación al caso del tiempo y sus relaciones con el calendario gregoriano.

1.1. Definición de granularidad

Una **granularidad** g en la recta real es una sucesión infinita monótona estrictamente creciente de números reales

$$g = \{x_k \in \mathbb{R}\}_{k\in\mathbb{Z}} \perp x_k < x_{k+1} \forall k \in \mathbb{Z} \tag{1}$$

Por ser siempre creciente estrictamente se tiene que necesariamente es no acotada ni por la izquierda ni por la derecha, más concretamente se cumple

$$\lim_{k\to-\infty} x_k = -\infty \wedge \lim_{k\to\infty} x_k = \infty \tag{2}$$

Dada la granularidad g definida anteriormente se define el operador

$$k \bullet g = x_k \forall k \in \mathbb{Z} \tag{3}$$

que no es sino otra forma de escribir la sucesión de forma que se haga referecia explícita a la granularidad, un mero truco notacional.

1.2. Ordenación de granularidades

Se dirá que una sucesión g *es más fina que* otra h y escribiremos $g \supset h$ ó $h \subset g$, o bien que h *es más gruesa que* g y escribiremos $h > g$ ó $g < h$, si la sucesión que define a h está incluida en la que define a g, es decir

$$\forall\, n \in \mathbb{Z}\, \exists\, m \in \mathbb{Z} \perp m \bullet g = n \bullet h \tag{4}$$

A la operación que devuelve ese entero m la llamaremos *cociente granular* y la denotaremos con el símbolo de la inversión sobre la granularidad

$$m = g^{-1}\,(n \bullet h) \Leftrightarrow m \bullet g = n \bullet h \quad \forall n \in \mathbb{Z}, \forall k \in \mathbb{Z},\ h > g \tag{5}$$

Se define la granularidad continua g_∞ como el límite superior de todas las granularidades en orden de más gruesa a más fina.

1.3. El calendario gregoriano

Cuando la magnitud física que representa la recta real es el tiempo se hablará de granularidades temporales. Estas se suelen definir mediante cualquier regla de cálculo de la sucesión $\{k \bullet g\}_{k\in\mathbb{Z}}$, como puede ser dar un punto de partida $0 \bullet g$ y una regla de sucesión temporal que permita reconstruir la sucesión de forma recursiva

$$(m+1) \bullet g = Succ\,(m \bullet g) \tag{6}$$

A las granularidades temporales correspondientes a las unidades básicas de tiempo en el **calendario gregoriano**[1] les llamaremos *granularidades temporales básicas* y son las correspondientes

[1] En 1582 fue cuando se inició el calendario gregoriano por mandato del Papa Gregorio XIII, corrigiendo el anterior calendario juliano que había provocado un desfase de 10 días con respecto al ciclo anual, y que fueron eliminados del calendario por decreto, pasando directamente del 4 al 15 de octubre y dando lugar a cosas tan curiosas como que la gente protestara en grandes revueltas creyendo que se le habían restado 10 días de vida, o que Santa Teresa muriera el día 4 y fuera enterrada el día 15, es decir, al día siguiente. Todos los países no adoptaron la nueva regla al unísono; esta es la razón de que Cervantes y Shakespeare muriesen formalmente en la misma fecha, el 23 de abril de 1616, pero con diez días de diferencia. Pero en cualquier caso se tomará el 1583 como primer año del calendario gregoriano. El calendario juliano introducía el concepto de año bisiesto con un día más, el 29 de febrero, cada 4 años. El gregoriano mejoraba esa aproximación eliminando un bisiesto cada siglo que no fuera múltiplo de 4.

a las sucesiones de los años, los meses, los días, las horas, los minutos, y los segundos. Es posible extender muy facilmente esta definición a granularidades más finas como la milésima de segundo, o el nanosegundo pero por razones técnicas la presente implementación se limita al segundo. De esta forma se define la base gregoriana de granularidades dispuestas en orden de más gruesa a más fina

$$\mathcal{G}_6 = \left\{ \begin{array}{cc} g_{\mathbf{y}}: & \textbf{anual} \\ g_{\mathbf{m}}: & \textbf{mensual} \\ g_{\mathbf{d}}: & \textbf{diaria} \\ g_{\mathbf{h}}: & \textbf{horaria} \\ g_{\mathbf{i}}: & \textbf{minutal} \\ g_{\mathbf{s}}: & \textbf{secundal} \end{array} \right\} \tag{7}$$

Al conjunto $\mathcal{IG}_6 = \{\mathbf{y}, \mathbf{m}, \mathbf{d}, \mathbf{h}, \mathbf{i}, \mathbf{s}\}$ le llamaremos índice de la base gregoriana de granularidades.

Así, la granularidad del segundo es más fina que la del minuto y que todas las demás y la del año es la más gruesa: $g_{\mathbf{y}} > g_{\mathbf{m}} > g_{\mathbf{d}} > g_{\mathbf{h}} > g_{\mathbf{i}} > g_{\mathbf{s}}$. Se recalca esta cuestión de orden para que no se confunda el lector en lo que resta del documento.

Para la granularidad diaria $\{k \bullet g_{\mathbf{d}}\}_{k\in\mathbb{Z}}$ sus elementos son el número de días transcurridos desde un punto de origen común a todas las granularidades $0 \bullet g_r = 0\ \forall r \in \mathcal{IG}_6$, que se suele tomar en muchas aplicaciones, bases de datos y hojas de cálculo como el IT que comienza el 1 de enero de 1900 a las 0 horas, 0 minutos y 0 segundos. Nótese que las granularidades que van del día al segundo son regulares en el sentido de que los intervalos entre sus elementos tienen igual longitud dentro de cada una, es decir,

$$\exists \lambda_r \in \mathbb{R}^+ \perp (k+1) \bullet g_r - k \bullet g_r = \lambda_r \Rightarrow k \bullet g_r = k \times \lambda_r \forall k \in \mathbb{Z},\ r = \mathbf{d}, \mathbf{h}, \mathbf{i}, \mathbf{s} \tag{8}$$

La unidad del calendario gregoriano es el día y por lo tanto

$$\begin{array}{l} \lambda_{\mathbf{d}} = 1 \\ \lambda_{\mathbf{h}} = \frac{1}{24} \\ \lambda_{\mathbf{i}} = \frac{1}{24\times 60} = \frac{1}{1440} \\ \lambda_{\mathbf{s}} = \frac{1}{24\times 60\times 60} = \frac{1}{86400} \end{array} \tag{9}$$

También se pueden definir otras granularidades como la semanal $g_{\mathbf{w}}$, que es regular con $\lambda_{\mathbf{w}} = 7$, aunque no formarán parte de la base gregoriana.

Sin embargo las granularidades mensual y la anual son irregulares pues hay meses más largos que otros y lo mismo pasa con los años y sólo se tienen los valores aproximados de las periodicidades $\lambda_{\mathbf{y}} \approx 365{,}2425$ y $\lambda_{\mathbf{m}} \approx 30{,}436875$.

Para la granularidad mensual $k \bullet g_{\mathbf{m}}$ son los días transcurridos tras los n primeros meses desde el punto de origen si $k \geq 0$, o bien los días que faltan hasta ese punto contando desde $-n$ meses antes si es $k < 0$.

Análogamente, $k \bullet g_{\mathbf{y}}$ son los días transcurridos tras los k primeros años desde el punto de origen si $k \geq 0$ o bien los días que faltan hasta ese punto contando desde $-k$ años antes si es $k < 0$.

Existen métodos sencillos para la determinación de $k \bullet g_{\mathbf{m}}$ y $k \bullet g_{\mathbf{y}}$ a partir de la regla de cálculo de los años bisiestos que son básicamente los múltiplos de 4 que no lo son de 100 a no ser que sean de 400. Esto da un error en el ciclo anual menor de 3 días cada 10000 años, por lo que se puede dar por válido al menos hasta el año 1582+3333=4915.

A efectos de implementación se usará la librería open-source libtai `http://cr.yp.to/libtai.html` para el almacenamiento y manipulación de fechas y tiempos.

2. Instantes de Tiempo

Los instantes de tiempo, pueden medirse en un rango de unidades que va de años a segundos. Estas unidades representan la granularidad de cada uno de ellos. Un instante de tiempo se define tomando valores en cada uno de los conceptos para los cuales existe una granularidad: año, mes, día, hora, minuto y segundo. Pueden omitirse secuencialmente de derecha a izquierda sin que falte nunca el último de todos, el año. Dos instantes de tiempo guardan una relación de orden entre sí. Por otro lado, un instante de tiempo puede estar contenido dentro de otro.

A continuación se dará contenido formal matemático a estas ideas intuitivas.

2.1. Definición de Instante de Tiempo

Desde el punto de vista matemático un **instante de tiempo continuo** o simplemente instante de tiempo, a partir de ahora **IT**, es un intervalo de Borel de la recta real de la forma

$$\mathcal{I}(k, g_r) = [k \bullet g_r, (k+1) \bullet g_r) \subset \mathbb{R} \wedge r \in \mathcal{IG}_6 \tag{10}$$

donde $k \in \mathbb{Z}$ es un número entero cualquiera, g_r es una granularidad básica gregoriana, $k \bullet g_r \in \mathbb{R}$ representa el punto de inicio del IT y $(k+1) \bullet g_r \in \mathbb{R}$ su límite superior, que no le pertenece propiamente, por ser un intervalo abierto por la derecha.

He aquí algunos ejemplos válidos de IT

- El día 9 de febrero de 1900 es el IT $\mathcal{I}(40, g_{\mathbf{d}}) = [40, 41)$
- El mes de enero de 1900 es el IT $\mathcal{I}(0, g_{\mathbf{m}}) = [0, 31)$

Al conjunto de todos los IT para cualquiera de estas 6 granularidades le llamaremos $\mathfrak{C}_6$. La cardinalidad de cada $\mathfrak{C}_{R>6}$ definible por extensión de la base de granularidades sería la del infinito numerable ∞. Sólo en el límite $\mathfrak{C}_\infty$ se alcanzaría la cardinalidad del continuo ∞^∞.

2.1.1. Ampliación de $\mathfrak{C}_6$ con instantes de tiempo impropios

El conjunto de los IT de una misma granularidad es una teselación de la recta real, es decir, la intersección de cada par de ellos es nula y su unión es toda la recta real. Cada uno de ellos define por tanto una sucesión infinita en la cual es posible definir los instantes de tiempo límite como los límites a los que convergen, para todas las granularidades, las sucesiones de instantes por la izquierda y por la derecha

$$\begin{array}{c} \mathcal{I}(-\infty) = \text{lím}_{k \to -\infty} \mathcal{I}(k, g) = \{-\infty\} \, \forall g \in \mathcal{G}_6 \\ \mathcal{I}(\infty) = \text{lím}_{k \to \infty} \mathcal{I}(k, g) = \{\infty\} \, \forall g \in \mathcal{G}_6 \end{array} \tag{11}$$

Obsérvese que dichos instantes límite no son propiamente intervalos sino que se trata de puntos de la recta real ampliada con los que se amplía a su vez el conjunto $\mathfrak{C}_6$.

También se amplía el concepto de IT con el IT desconocido o indeterminado $\mathcal{I}(?)$ que se utilizará para expresar que no existe el resultado de una expresión que debería devolver un IT. Este IT indeterminado no pertenece realmente a $\mathfrak{C}_6$ pero en ocasiones se escribirá $\mathfrak{C}_6 - \{\mathcal{I}(?)\}$ para recalcarlo. Del mismo modo se escribirá $\mathfrak{C}_6 - \{\mathcal{I}(-\infty), \mathcal{I}(\infty)\}$ o cualquier combinación de exclusiones. A estos tres IT's se les llamará IT's impropios y al resto, los que se pueden expresar como $\mathcal{I}(k, g)$, se les llamará IT propios. Si no se indica otra cosa se entenderá siempre que un IT es propio.

2.2. Estructura Algebraica de $\mathfrak{C}_6$

Se van a definir operaciones algebraicas entre instantes de tiempo y entre estos y los números enteros que permitirán manejar estos objetos desde un punto vista formal.

2.2.1. Operaciones externas

A continuación se describen algunas funciones de acceso a las propiedades de un IT y otras funciones relacionadas con el tratamiento del tiempo

Granularidad A la sucesión g le llamaremos granularidad del IT, y a la función que la devuelve

$$granul\left(\mathcal{I}\left(k,g\right)\right)=g\in\mathcal{G}_6\quad\forall k\in\mathbb{Z}\tag{12}$$

Ordinal Al valor entero k le llamaremos ordinal del IT y a la función correspondiente

$$ord\left(\mathcal{I}\left(k,g\right)\right)=k\in\mathbb{Z}\quad\forall g\in\mathcal{G}_6\tag{13}$$

Punto de inicio Al valor real $t=k\bullet g$ le llamaremos punto de inicio, y a la función que devuelve dicho punto de inicio para un IT le llamaremos

$$startOf\left(\mathcal{I}\left(k,g\right)\right)=t=k\bullet g\in\mathbb{R}\quad\forall k\in\mathbb{Z}\,g\in\mathcal{G}_6\tag{14}$$

Distancia en una granularidad Se define la función distancia entre dos puntos reales en una granularidad $g_r\in\mathcal{G}_6$ como el número de IT's de esa granularidad incluidos entre esos dos puntos. Es decir

$$\left\langle t,\,t'\right\rangle_{g_r}=Card\left\{\mathcal{I}\left(j,g_r\right)\middle|\,t\leq j\bullet g_r\wedge\left(j+1\right)\bullet g_r<t'\right\}\quad\forall t,t'\in\mathbb{R}\tag{15}$$

Se define la distancia entre dos IT's en una granularidad como la distancia entre sus puntos de origen en esa misma granularidad.

$$\left\langle\mathcal{I}\left(k,g\right),\,\mathcal{I}\left(k',g'\right)\right\rangle_{g_r}=\left\langle k\bullet g,\,k'\bullet g'\right\rangle_{g_r}\quad\forall k,k'\in\mathbb{Z}\tag{16}$$

En particular para dos IT's con granularidad g_r se tiene que

$$\left\langle\mathcal{I}\left(k,g_r\right),\,\mathcal{I}\left(k',g_r\right)\right\rangle_{g_r}=\left\langle k\bullet g_r,\,k'\bullet g_r\right\rangle_{g_r}=\left\{\begin{array}{cc}k'-k & si\,k'>k\\ 0 & si\,k'\leq k\end{array}\right.\quad\forall k,k'\in\mathbb{Z}\tag{17}$$

Obviamente, cuanto más fina es la granularidad mayor es la distancia entre los mismos puntos.

Funciones de coordenadas gregorianas En virtud del calendario gregoriano se definen también las funciones de coordenadas gregorianas

$$\begin{array}{cccc}year\left(t\right) & :\mathbb{R}\rightarrow & RG_{\mathbf{y}}=\mathbb{Z}\cap & [1583,\,4999999]\\ month\left(t\right) & :\mathbb{R}\rightarrow & RG_{\mathbf{m}}=\mathbb{Z}\cap & [1,\,12]\\ day\left(t\right) & :\mathbb{R}\rightarrow & RG_{\mathbf{d}}=\mathbb{Z}\cap & [1,\,31]\\ hour\left(t\right) & :\mathbb{R}\rightarrow & RG_{\mathbf{h}}=\mathbb{Z}\cap & [0,\,23]\\ minute\left(t\right) & :\mathbb{R}\rightarrow & RG_{\mathbf{i}}=\mathbb{Z}\cap & [0,\,59]\\ second\left(t\right) & :\mathbb{R}\rightarrow & RG_{\mathbf{s}}=\mathbb{Z}\cap & [0,\,59]\end{array}\tag{18}$$

que devuelven para cada punto de la recta real el año, mes, día, hora, minuto y segundo al que corresponde en dicho calendario. A los conjuntos RG_r les llamaremos *rangos de coordenadas gregorianas*. También se define del mismo modo la función que devuelve el día de la semana que, aunque no es una coordenada de la base tiene mucha importancia en el tratamiento del tiempo social.

$$weekDay\left(t\right):\mathbb{R}\rightarrow RG_{\mathbf{w}}=\mathbb{Z}\cap[1,\,7]\tag{19}$$

Representación de instantes de tiempo Un IT $\mathcal{I}(k, g_r)$ se puede representar como una r-tupla con $r = 1 \ldots 6$

$$IT(c_1, ..., c_r) : RG_1 \times \cdots \times RG_r \to \mathfrak{C}_6 \tag{20}$$

donde los $c_{k=1..s}$ son las coordenadas gregorianas del punto de inicio del IT incluyendo sólo las significativas en su granularidad.

$$\begin{array}{rlrl} c_1 = & year(k \bullet g_r) & c_2 = & month(k \bullet g_r) \\ c_3 = & day(k \bullet g_r) & c_4 = & hour(k \bullet g_r) \\ c_5 = & minute(k \bullet g_r) & c_6 = & second(k \bullet g_r) \end{array} \tag{21}$$

La función de representación $IT(c_1, ..., c_s)$ también se puede expresar como

$$IT(c_1, ..., c_6) = y\langle c_1\rangle\, m\langle c_2\rangle\, d\langle c_3\rangle\, h\langle c_4\rangle\, i\langle c_5\rangle\, s\langle c_6\rangle \tag{22}$$

Por ejemplo:

- $it_1 = IT(2005) = y2005$ será el IT que representa al año 2005, su granularidad será "anual".
- $it_2 = IT(2005, 5) = y2005m5$ será el IT que representa el mes 5 del año 2005, su granularidad será "mensual".
- $it_3 = IT(2005, 5, 15) = y2005m5d15$ será el IT que representa el día 15 de Mayo de 2005, su granularidad será "diaria".
- $it_4 = IT(2005, 5, 15, 0) = y2005m5d15h0$ será el IT que representa la hora 0 del día 15 de Mayo de 2005, su granularidad será "horaria".
- $it_5 = IT(2005, 5, 15, 0, 53) = y2005m5d15h0i53$ será el IT que representa el minuto 53 de la hora 0 del día 15 de Mayo de 2005, su granularidad será "minutal".
- $it_6 = IT(2005, 5, 15, 0, 53, 12) = y2005m5d15h0i53s12$ será el IT que representa el segundo 12 del minuto 53 de la hora 0 del día 15 de Mayo de 2005, su granularidad será "secundal".

2.2.2. Relación de Equivalencia

Se define la siguiente relación binaria entre instantes de tiempo

$$\mathcal{I}(k, g) = \mathcal{I}(k', g') \Leftrightarrow k = k' \wedge g = g' \tag{23}$$

Es decir dos instantes de tiempo son iguales si y sólo si tienen el mismo origen y la misma granularidad. Obviamente se trata de una *relación de equivalencia*, es decir, cumple las propiedades *reflexiva*, *simétrica* y *transitiva*.

2.2.3. Operadores siguiente y anterior de un IT

El **IT siguiente de un IT** es el instante de tiempo inmediatamente superior en la granularidad del IT, lo cual se puede expresar formalmente como

$$\mathcal{I}(k, g) + 1 = \mathcal{I}(k + 1, g) = [(k + 1) \bullet g, (k + 2) \bullet g)$$

Del mismo modo el **IT anterior de un IT** es el instante de tiempo inmediatamente inferior en la granularidad del IT, lo cual se puede expresar formalmente como

$$\mathcal{I}(k, g) - 1 = \mathcal{I}(k - 1, g) = [(k - 1) \bullet g, k \bullet g) \tag{24}$$

Estas operaciones se pueden aplicar recursivamente

$$
\begin{array}{c}
\mathcal{I}(k,g)+n=\mathcal{I}(k+n,g)=[(k+n)\bullet g,(k+n+1)\bullet g)\,\forall n\in\mathbb{Z}^{+}\\
\mathcal{I}(k,g)-n=\mathcal{I}(k-n,g)=[(k-n)\bullet g,(k-n+1)\bullet g)\,\forall n\in\mathbb{Z}^{+}\\
\mathcal{I}(k,g)+0=\mathcal{I}(k,g)-0=\mathcal{I}(k+0,g)=\mathcal{I}(k,g)
\end{array}
\tag{25}
$$

Por lo tanto se puede extender el concepto de **IT siguiente n-ésimo de un IT** a todo $\mathbb{Z}$. Obsérvese que esta operación es neutra respecto a los IT impropios

$$
\begin{array}{c}
\mathcal{I}(-\infty)+n=\mathcal{I}(-\infty)\,\forall n\in\mathbb{Z}\\
\mathcal{I}(\infty)+n=\mathcal{I}(\infty)\,\forall n\in\mathbb{Z}\\
\mathcal{I}(?)+n=\mathcal{I}(?)\,\forall n\in\mathbb{Z}
\end{array}
\tag{26}
$$

He aquí algunos ejemplos ilustrativos:
$y2005+1=y2006$
$y2005-1=y2004$
$y2005m3-10=y2004m5$
$y2006m9d4h23+2=y2006m9d5h1$

2.2.4. Relación de Orden

Se define la siguiente relación binaria entre instantes de tiempo

$$
\mathcal{I}(k,g)<\mathcal{I}(k',g')\Leftrightarrow(k\bullet g<k'\bullet g')\vee(k\bullet g=k'\bullet g'\wedge g<g')
\tag{27}
$$

que se lee así: un IT *es estrictamente menor que* otro si y sólo sí, o bien comienza antes, o bien comienza en el mismo momento pero acaba antes. Por ejemplo,$y2005<y2006$, $y2005m1<y2005$

En virtud de dicha relación se puede definir la relación binaria

$$
\mathcal{I}(k,g)\leq\mathcal{I}(k',g')\Leftrightarrow\mathcal{I}(k,g)<\mathcal{I}(k',g')\vee\mathcal{I}(k,g)=\mathcal{I}(k',g')
\tag{28}
$$

Esta relación cumple las propiedades de una *relación de orden total*

- *Reflexiva*: todo IT es menor o igual que sí mismo,

$$
\mathcal{I}(k,g)\leq\mathcal{I}(k,g)
\tag{29}
$$

- *Antisimétrica*: si un IT es menor o igual que otro y viceversa entonces son iguales

$$
\mathcal{I}(k,g)\leq\mathcal{I}(k',g')\wedge\mathcal{I}(k',g')\leq\mathcal{I}(k,g)\Rightarrow\mathcal{I}(k,g)=\mathcal{I}(k',g')
\tag{30}
$$

- *Transitiva*: si un IT es menor o gual que otro y éste lo es respecto a un tercero entonces el primero es menor o igual que este último

$$
\mathcal{I}(k,g)\leq\mathcal{I}(k',g')\wedge\mathcal{I}(k',g')\leq\mathcal{I}(k'',g'')\Rightarrow\mathcal{I}(k,g)=\mathcal{I}(k'',g'')
\tag{31}
$$

- *Conexa*: dados dos IT cualesquiera uno de los dos es menor o igual que el otro

$$
\mathcal{I}(k,g)\leq\mathcal{I}(k',g')\vee\mathcal{I}(k',g')\leq\mathcal{I}(k,g)\,\forall\mathcal{I}(k,g),\mathcal{I}(k',g')\in\mathfrak{C}_6
\tag{32}
$$

Por lo tanto es posible definir de la forma usual las funciones típicas de un orden total como el mínimo y el máximo de un conjunto de IT's o el límite por la izquierda y por la derecha.

Obsérvese que no es posible comparar un IT con $\mathcal{I}(?)$ y que

$$
\mathcal{I}(-\infty)<\mathcal{I}(k,g)<\mathcal{I}(\infty)\,\forall\mathcal{I}(k,g)\in\mathfrak{C}_6
\tag{33}
$$

2.2.5. Relación de Inclusión

Se define la siguiente relación binaria de inclusión entre instantes de tiempo como

$$it' \subset it \Leftrightarrow startOf(it) \leqslant startOf(it') \wedge startOf(Succ(it)) \geqslant startOf(Succ(it')) \tag{34}$$

que se lee así: un IT está incluido en otro si no comienza antes ni termina después. En la notación usada anteriormente se expresaría así

$$it' = \mathcal{I}(k', g') \subset \mathcal{I}(k, g) = it \Leftrightarrow k \bullet g \leq k' \bullet g' \wedge (k' + 1) \bullet g' \leq (k + 1) \bullet g \tag{35}$$

Por ejemplo $y2005m1 \subset y2005$, $y2005m1d1h12 \subset y2005m1d1$, $y2005m1d1h12i6 \subset y2005m1$.

Obsérvese que la inclusión no está definida si interviene $\mathcal{I}(?)$ y que tanto $\mathcal{I}(-\infty)$ como $\mathcal{I}(\infty)$ ni incluyen ni están incluidos en ningún otro IT.

Se define también la relación de inclusión estricta

$$it' \subsetneq it \Leftrightarrow it' \subset it \wedge it \neq it' \tag{36}$$

Esta relación cumple las propiedades de una relación de *orden parcial* pero no un orden total, pues claramente no es conexa, ya que puede ocurrir que un instante temporal no esté incluido en otro ni éste en aquel. En tal caso, uno de ellos será estrictamente menor que el otro, ya que lo que no puede haber es instantes temporales solapados parcialmente, debido a la propia definición de la base de granularidades gregoriana.

2.2.6. Interior de un IT

Para cada IT se puede definir la clase de todos los IT's que están incluidos en ese IT, a la que llamaremos **interior de un IT**

$$Inside\,(it) = \{it' \in \mathfrak{C}_6\,|it' \subset it\} \tag{37}$$

Por ejemplo

$$Inside\,(y2005m1d1h12i6) = \{y2005m1d1h5i6,\ y2005m1d1h5i6s0,\ ...,\ y2005m1d1h5i6s59\}$$

Obviamente, el interior de un IT contiene el interior de cada uno de sus elementos y por lo tanto los elementos del interior de un IT tienen todas las granularidades menos gruesas que la de ese IT, excepto él mismo, claro está, que tiene la granularidad más gruesa de todos.

$$\begin{array}{c} \mathcal{I}(k', g') \in Inside\,(\mathcal{I}(k, g)) \Rightarrow Inside\,(\mathcal{I}(k', g')) \subset Inside\,(\mathcal{I}(k, g)) \\ \mathcal{I}(k', g') \in Inside\,(\mathcal{I}(k, g)) - \{\mathcal{I}(k, g)\} \Rightarrow g' < g \\ g' \leq g \Rightarrow \exists \mathcal{I}(k', g') \in Inside\,(\mathcal{I}(k, g)) \end{array} \tag{38}$$

Como corolario se puede extraer que el interior de un IT tiene al menos un elemento de granularidad secundal.

2.2.7. Contenedores de un IT

Para cada IT $\mathcal{I}(k, h)$ de granularidad $h < g$ existe un único IT de granularidad g que lo incluye, pongamos $\mathcal{I}(l, g)$, al que llamaremos *contenedor de* $\mathcal{I}(k, h)$ *en la granularidad* $g > h$ y se escribirá

$$\mathcal{I}(l, g) = g^{-1}\mathcal{I}(k, h) \Leftrightarrow \mathcal{I}(k, h) \subset \mathcal{I}(l, g) \Leftrightarrow k \in [a, b] \cap \mathbb{Z}\ \forall h < g \tag{39}$$

donde a los valores límite a y b de la expresión anterior les llamaremos ***límites superior e inferior de*** $\mathcal{I}(l, g)$ ***en la granularidad*** $h < g$. Según se puede ver en la ecuación 5 en la página 3 se pueden calcular mediante inversiones granulares de los puntos inicial y final de $\mathcal{I}(l, g)$

$$\begin{array}{l} a = \liminf_h (\mathcal{I}(l, g)) = h^{-1}(l \bullet g) \\ b = \limsup_h (\mathcal{I}(l, g)) = h^{-1}((l + 1) \bullet g) - 1 \end{array} \quad \forall h < g \tag{40}$$

3. Conjuntos Temporales

Un **conjunto temporal**, en adelante **CT**, es un tipo especial de conjunto de instantes de tiempo con cierta estructura y que cumple determinadas normas que se explican a continuación. Más adelante se exponen algunas propiedades importantes así como algunas operaciones algebraicas que se pueden definir entre CT's y también otras que relacionan CT's e IT's.

3.1. Definición y propiedades de un CT

Un CT es un conjunto de instantes de tiempo propios de cualquier granularidad básica que incluye el interior de cada uno de sus elementos.

A la familia de todos los conjuntos temporales le llamaremos $\mathcal{CT}_6$ y se tendrá

$$CT \in \mathcal{CT}_6 \Leftrightarrow CT \subset \mathfrak{C}_6 - \{\mathcal{I}(?), \mathcal{I}(-\infty)\mathcal{I}(\infty)\} \ \wedge \ Inside\,(it) \subset CT\ \forall it \in CT \tag{41}$$

La cardinalidad de $\mathcal{CT}_6$ será por tanto la misma que la del conjunto $\mathcal{P}(\mathfrak{C}_6)$ de las partes de $\mathfrak{C}_6$, es decir, la del continuo ∞^∞.

3.1.1. Sucesiones disjuntas de IT's

Llamaremos *sucesión disjunta de IT's* a una sucesión de IT's tales que ninguno de ellos está incluido en ningún otro. Puesto que los IT's no se pueden solapar parcialmente es lo mismo que decir que todos sus elementos son disjuntos. Es decir una sucesión disjunta es de la forma

$$S = \{it_j\}_{j \in J \subset \mathbb{Z}} \perp it_j \subsetneq it_{j'} \forall j \neq j' \wedge j,\, j' \in J \tag{42}$$

donde J es el conjunto de índices enteros contiguos de la sucesión que tiene efecto puramente nominal, es decir, dos sucesiones disjuntas se considerarán iguales, si y sólo si ambas tienen exactamente los mismos elementos, independientemente de que sus conjuntos de índices sean iguales, o bien sea el uno cualquier translación del otro. Al conjunto de todas las sucesiones disjuntas de IT's le llamaremos $\mathcal{SD}_6$.

Se define la *suma disjunta de dos sucesiones disjuntas de IT's* y se representará mediante el operador $S \oplus S'$ como la sucesión disjunta resultante de la unión de ambas sucesiones, eliminando de cada una todos aquellos IT's que estén incluidos estrictamente en algún elemento de la otra sucesión.

$$\begin{gathered} S \oplus S' = \{it \in S |\, \nexists it' \in S' \perp it \subsetneq it'\} \cup \{it' \in S' |\, \nexists it \in S \perp it' \subsetneq it\} \\ S \oplus S' \in \mathcal{SD}_6\ \forall S,\ S' \in \mathcal{SD}_6 \end{gathered} \tag{43}$$

Se define el *producto disjunto de dos sucesiones disjuntas de IT's* como su propia intersección y se representará mediante el operador $S_1 \otimes S_2 = S_1 \cap S_2$. Como ningún elemento de S_1 incluye a ningun otro de S_1 y ningún elemento de S_2 incluye a ninguno de S_2, los que pertenezcan a ambos a la vez tampoco incluirán a ningún otro de la intersección.

$$S \otimes S' = S_1 \cap S_2 \in \mathcal{SD}_6\ \forall S,\ S' \in \mathcal{SD}_6 \tag{44}$$

Se define la *diferencia disjunta de dos sucesiones disjuntas de IT's* y se representará mediante el operador $S_1 \ominus S_2$ como la sucesión disjunta de los elementos de S_1 que no incluyen estrictamente a ningún elemento de S_2.

$$S \ominus S' = \{it \in S |\, \nexists it' \in S' \perp it' \subsetneq it\} \in \mathcal{SD}_6\ \forall S,\ S' \in \mathcal{SD}_6 \tag{45}$$

Llamaremos *conjunto temporal generado por una sucesión disjunta* a la unión de los interiores de los elementos de esa sucesión

$$\underbrace{S} = \bigcup_{it \in S} Inside\,(it) \tag{46}$$

3.1.2. Sucesión maximal de un CT

Para todo elemento IT de un CT existe un elemento máximo de CT que le contiene, al que llamaremos *elemento maximal de IT en CT*

$$Maximal\,(it, CT) = it' \Leftrightarrow it'' \subset it' \in CT\, \forall\, it \subset it'' \in CT \tag{47}$$

Se dice que un IT de un CT es un *elemento maximal* de ése CT si es su propio maximal en CT, es decir, si no pertenece al interior de ningún otro elemento de ese CT. A la sucesión disjunta, finita o infinita, de los elementos maximales de un CT, ordenada de menor a mayor, le llamaremos *sucesión maximal* de ese CT

$$Maximal\,(CT) = \widehat{CT} = \{it_j\}_{j\in J\subset\mathbb{Z}} \wedge \left\{ \begin{array}{c} it_j = Maximal\,(it_j, CT) \\ it_j < it_{j+1} \end{array} \right. \quad \forall j,\, j+1 \in J \tag{48}$$

Los interiores de dos elementos maximales distintos de un mismo CT son conjuntos disjuntos, es decir

$$it \neq it' \Leftrightarrow Inside\,(it) \cap Inside\,(it') = \emptyset\; \forall it, it' \in \widehat{CT} \tag{49}$$

Por lo tanto, todo CT es el *conjunto temporal generado por su sucesión maximal*

$$CT = \underbrace{\widehat{CT}} = \bigcup_{it\in\widehat{CT}} Inside\,(it) \tag{50}$$

El **sucesor maximal de un IT en un CT** se define como el primer elemento maximal del CT que es estrictamente posterior a dicho IT. Análogamente, el **predecesor maximal de un IT en un CT** se define como el último elemento maximal del CT que es estrictamente anterior a dicho IT.

$$\begin{array}{l} it + \widehat{CT} = it' \Leftrightarrow it' \in \widehat{CT} \wedge it < it' \leq it''\; \forall it'' \in \widehat{CT} \\ it - \widehat{CT} = it' \Leftrightarrow it' \in \widehat{CT} \wedge it > it' \geq it''\; \forall it'' \in \widehat{CT} \end{array} \tag{51}$$

3.1.3. Sucesor y predecesor de un IT en un CT

Se define el punto *inmediatamente superior a un IT en un CT*, o **supremo de un IT en un CT**, como el menor valor real posterior al punto final de ese IT tal que existe algún elemento del CT cuyo punto de inicio sea ese valor. De forma análoga se define el punto *inmediatamente inferior a un IT en un CT*, o **ínfimo de un IT en un CT**, como el mayor valor real estrictamente menor que el punto inicial de ese IT tal que existe algún elemento del CT cuyo punto final sea ese valor.

$$\begin{array}{l} \sup\left(\mathcal{I}\left(k,g\right), CT\right) = \text{mín}\left\{t \in \mathbb{R} \middle| \exists \mathcal{I}\left(k',g'\right) \in CT \perp (k+1) \bullet g \leq t = k' \bullet g'\right\} \\ \inf\left(\mathcal{I}\left(k,g\right), CT\right) = \text{máx}\left\{t \in \mathbb{R} \middle| \exists \mathcal{I}\left(k',g'\right) \in CT \perp k \bullet g > t = (k'+1) \bullet g'\right\} \end{array} \tag{52}$$

Al subconjunto de CT de los elementos cuyo origen es el supremo de un IT en ese CT se le llamará **subconjunto supremo de IT en CT**. De forma análoga, al subconjunto de CT de los elementos cuyo origen es el ínfimo de un IT en ese CT se le llamará **subconjunto ínfimo de IT en CT**.

$$\begin{array}{r} cSup\left(\mathcal{I}\left(k,g\right), CT\right) = \left\{\mathcal{I}\left(k',g'\right) \in CT \middle| k' \bullet g' = \sup\left(\mathcal{I}\left(k,g\right), CT\right)\right\} \\ cInf\left(\mathcal{I}\left(k,g\right), CT\right) = \left\{\mathcal{I}\left(k',g'\right) \in CT \middle| k' \bullet g' = \inf\left(\mathcal{I}\left(k,g\right), CT\right)\right\} \end{array} \tag{53}$$

El subconjunto supremo de un IT en un CT es el interior del sucesor maximal de IT en CT. Del mismo modo, el subconjunto ínfimo de un IT en un CT es el interior del predecesor maximal de IT en CT.

$$
\begin{array}{c}
cSup\left(\mathcal{I}\left(k,g\right),CT\right)=Inside\left(\mathcal{I}\left(k,g\right)+\widehat{CT}\right)\\
cInf\left(\mathcal{I}\left(k,g\right),CT\right)=Inside\left(\mathcal{I}\left(k,g\right)-\widehat{CT}\right)
\end{array}
\tag{54}
$$

El **sucesor de un IT en un CT** se define como el elemento del subconjunto supremo de IT en CT que tiene la máxima granularidad no superior a la del propio IT. El **predecesor de un IT en un CT** se define como el elemento del subconjunto ínfimo de IT en CT que tiene máxima granularidad no superior a la del propio IT.

$$
\begin{array}{c}
\mathcal{I}\left(k,g\right)+CT=\mathcal{I}\left(k',g'\right)\Leftrightarrow\mathcal{I}\left(k',g'\right)\in X=cSup\left(\mathcal{I}\left(k,g\right),CT\right)\wedge g''<g'\leq g\forall\mathcal{I}\left(k'',g''\right)\in X\\
\mathcal{I}\left(k,g\right)-CT=\mathcal{I}\left(k',g'\right)\Leftrightarrow\mathcal{I}\left(k',g'\right)\in Y=cInf\left(\mathcal{I}\left(k,g\right),CT\right)\wedge g''<g'\leq g\forall\mathcal{I}\left(k'',g''\right)\in Y
\end{array}
\tag{55}
$$

Tanto el sucesor como el predecesor de un IT en un CT, si existen, entonces son únicos por su propia definición, ya que no puede haber dos IT's distintos con la misma granularidad y el mismo origen en el supremo, ni tampoco con el mismo final en el ínfimo.

3.1.4. Sucesor n-ésimo de un IT en un CT

Se puede extender recursivamente la definición de **sucesor n-ésimo de un IT en un CT** del siguiente modo

$$
\begin{array}{c}
it+n\times CT=\left(it+\left(n-1\right)\times CT\right)+CT\ \forall n\in\mathbb{Z}^{+}\\
it-n\times CT=\left(it-\left(n-1\right)\times CT\right)-CT\ \forall n\in\mathbb{Z}^{+}
\end{array}
\tag{56}
$$

El caso de $n=0$ es realmente singular pues para que exista $it+0\times CT=it-0\times CT$ es necesario que se igualen las dos reglas de recursión anteriores

$$
\begin{array}{c}
it+0\times CT=\left(it-CT\right)+CT\\
it-0\times CT=\left(it+CT\right)-CT
\end{array}
\tag{57}
$$

Si $it\in CT$ entonces está claro que $\left(it-CT\right)+CT=it=\left(it+CT\right)-CT$ pero si $it\notin CT$ entonces el sucesor en CT del predecesor de it en CT es el mismo que el sucesor de it en CT, ya que no puede existir ningún elemento de CT entre ambos y por tanto

$$
\begin{array}{c}
\left(it+CT\right)=\left(it-CT\right)+CT\\
\left(it+CT\right)-CT<\left(it-CT\right)+CT
\end{array}
\tag{58}
$$

Así pues no se puede definir $it+0\times CT$ por lo que se ha de devolver $\mathcal{I}\left(?\right)$. Es decir

$$
it+0\times CT=\left\{\begin{array}{cc}
it & si\ it\in CT\\
\mathcal{I}\left(?\right) & si\ it\notin CT
\end{array}\right.
\tag{59}
$$

3.1.5. Granularidad de un CT

Cada elemento de un CT puede tener una granularidad distinta y todo CT tiene al menos un IT de granularidad secundal. Sin embargo nada le obliga a tener elementos de granularidad mayor que una dada. La granularidad de los elementos maximales de un CT conforman la sucesión de granularidades máximas que pueden tener sus elementos en cada intervalo de la recta real definido por cada IT maximal. Llamaremos *granularidad minimax de un CT* o simplemente *granularidad de un CT* a la menos gruesa de las granularidades de sus elementos maximales.

$$
granul\left(CT\right)=\min_{it\in\widehat{CT}}\left\{granul\left(it\right)\right\}
\tag{60}
$$

En determinadas circunstancias, cuando se calcula el sucesor de un IT en un conjunto, el conocer su granularidad puede ayudar a ahorrar trabajo en la búsqueda dentro del subconjunto supremo de ese IT en el CT. La información es particularmente útil en conjuntos temporales Ct un conjunto de granularidad g fija para todos sus maximales. En tal caso, para cada IT de granularidad $h < g$ se puede calcular el contenedor de $\mathcal{I}(k,h)$ de granularidad g, así como los límites inferior y superior de dicho contenedor, según se vio en las ecuaciones 39 y 40 en la página 9 y que se escriben aquí como funciones de (k,h,g)

$$\begin{array}{r}\mathcal{I}(l,g) = g^{-1}\mathcal{I}(k,h) \\ a(k,h,g) = \text{lím inf}_h(\mathcal{I}(l,g)) = h^{-1}(l \bullet g) \\ b(k,h,g) = \text{lím sup}_h(\mathcal{I}(l,g)) = h^{-1}((l+1) \bullet g) - 1\end{array} \tag{61}$$

Si $\mathcal{I}(k,h) \notin Ct$ entonces ningún elemento de granularidad h incluido en su contenedor de granularidad g puede pertenecer a Ct; y si $\mathcal{I}(k,h) \in Ct$ entonces todos los elementos de granularidad h incluidos en su contenedor de granularidad g pertenecen a Ct:

$$\mathcal{I}(j,h) \in Ct\, \forall a(k,h,g) \leq j \leq b(k,h,g) \Leftrightarrow \mathcal{I}(k,h) \in Ct \tag{62}$$

Por un lado, si $\mathcal{I}(k,h) \in Ct$ tenemos un método que reduce drásticamente el coste computacional del sucesor n-ésimo

$$\mathcal{I}(k,h) + n \times Ct = \mathcal{I}(k+n,h)\ \forall n = a-k, ..., b-k \tag{63}$$

Por otro lado, si $\mathcal{I}(k,h) \notin Ct$ el sucesor y el predecesor simples hay que buscarlos fuera de su contenedor de granularidad g

$$\begin{array}{c}\mathcal{I}(k,h) + Ct = \mathcal{I}(b(k,h,g),h) + Ct \\ \mathcal{I}(k,h) - Ct = \mathcal{I}(a(k,h,g),h) - Ct\end{array} \tag{64}$$

3.1.6. Fechados

Se llamará **fechado** a un conjunto temporal $F \in \mathcal{CT}_6$ no acotado inferior ni superiormente, es decir, que cumpla que

$$\forall t \in \mathbb{R}\, \exists \mathcal{I}(k,g), \mathcal{I}(k',g') \in F \perp k \bullet g < t < k' \bullet g' \tag{65}$$

Evidentemente, la sucesión maximal de un fechado no está acotada ni inferior ni superiormente. El sucesor n-ésimo de un IT finito $it \in \mathfrak{C}_6 - \{\mathcal{I}(-\infty), \mathcal{I}(\infty)\}$ también es finito, es decir

$$it + n \times F \notin \{\mathcal{I}(-\infty), \mathcal{I}(\infty)\}\, \forall n \in \mathbb{Z} \tag{66}$$

Los fechados tienen una gran importancia pues son la base para la creación de series temporales en tiempo continuo, así como para la definición de algunas operaciones importantes de CT's que se verán más adelante.

Aunque los instantes impropios $\mathcal{I}(-\infty)$, $\mathcal{I}(?)$, $\mathcal{I}(\infty)$ no pertenecen a ningún CT, a veces se obrará como si así fuera por comodidad notacional. Los instantes límite pueden pertenecer asintóticamente a un CT, y no así el instante indeterminado. En el caso de los fechados, los instantes límite $\mathcal{I}(-\infty)$, $\mathcal{I}(\infty)$ le pertenecen asintóticamente. Se puede hablar de semifechados por la izquierda o por la derecha para aquellos CT que estén acotados sólo por la izquierda o por la derecha, respectivamente.

3.1.7. Distancia en un CT

Se define la función distancia entre dos puntos reales en un CT como el número de IT's maximales de ese CT incluidos entre esos dos puntos. Es decir

$$\langle t, t' \rangle_{CT} = Card \left\{ \mathcal{I}(j,g) \in \widehat{CT} \middle| t \leq j \bullet g < (j+1) \bullet g < t' \right\} \quad \forall t, t' \in \mathbb{R} \tag{67}$$

Se define la distancia entre dos IT's en un CT como la distancia entre sus puntos de origen en ese mismo CT.

$$\langle \mathcal{I}(k,g), \mathcal{I}(k',g') \rangle_{CT} = \langle k \bullet g, k' \bullet g' \rangle_{CT} \quad \forall k, k' \in \mathbb{Z} \tag{68}$$

3.1.8. Notas para la implementación de CT's

Un CT se puede construir de varias formas:

- Por enumeración exhaustiva de sus elementos maximales, siempre que sea un conjunto finito.
- Mediante una función booleana explícita de pertenencia que devuelva *cierto* para cada IT que sea elemento suyo y *falso* para el resto.

$$Include\,(CT, it) = \left\{ \begin{array}{ll} 1 & si\,it \in CT \\ 0 & si\,it \notin CT \end{array} \right. \tag{69}$$

- Implícitamente, mediante un IT que le pertenezca y una regla de recursión reversible, es decir, que permita calcular cuál es el IT de ese conjunto inmediatamente superior (**sucesor**) o inferior (**predecesor**) a un IT cualquiera.

Sin embargo, será indispensable implementar al menos los métodos $Include\,(CT, it)$, $it + CT$ y $it - CT$ para cada clase de CT y cada posible granularidad de it de la forma más eficiente que sea posible aprovechando las características propias de cada uno, ya que estos métodos se utilizarán de forma masiva en conjuntos y series temporales. Siempre que sea posible y dé mayores prestaciones, también se implementará el método extendido $it+n\times CT$, ya que, aunque por defecto este método se puede definir por recursión de sucesores y predecesores, se usará tan a menudo que cualquier mejora producirá importantes ahorros en el tiempo de proceso.

3.2. CT finitos

Aquí se describen algunas clases de CT's finitos construidos de forma muy sencilla a partir de IT's

3.2.1. CT interior de un IT: `Inside`

Dado un IT cualquiera se define el CT interior de ese IT como ya se ha descrito en 70

$$Inside\,(it) = \{it' \in \mathfrak{C}_6\,|it' \subset it\} \tag{70}$$

3.2.2. CT generado por una lista de ITs: `SetOfCTimes`

Dada una lista finita cualquiera de IT's, se puede construir una sucesión disjunta ordenando previamente el conjunto y eliminando los elementos que estén incluidos en otros, construyendo así su sucesión maximal que servirá para optimizar los métodos de cálculo. A esta operación le llamaremos conjunto temporal generado por una lista finita de IT's

$$SetOfCTimes(\{it_j\}_{j=1,\ldots,n}) \;=\; \bigcup_{j=1}^{n} Inside\,(it_j) \tag{71}$$

3.2.3. CT intervalo entre dos IT's: `In`

Dados dos IT's a y b cualesquiera se define el CT intervalo como

$$[a, b] = In\,(a, b) = \{it \in \mathfrak{C}_6 \,|\, a \leq it \leq b\} \tag{72}$$

3.3. CT's constantes

Se trata de conjuntos temporales constantes en el sentido de que se definen sin necesidad de especificar ningún parámetro. La condición de incluir los interiores de todos sus elementos se desprende de las propias definiciones de forma trivial por lo que no se demuestra en cada caso. Todas estas constantes son fechados excepto el conjunto vacío.

En primer lugar se describen los fechados básicos completos, en el sentido de que recubren toda la recta real en las diferentes granularidades básicas. Después se tratan el conjunto vacío y el fechado básico pascual.

3.3.1. Fechado básico completo anual o conjunto universal

Llamaremos conjunto universal (cAll) o fechado anual (cYearly) al conjunto de todos los IT posibles que en esta implementación será el conjunto $\mathfrak{C}_6$

$$it \in cAll \; \forall it \in \mathfrak{C}_6 \tag{73}$$

Evidentemente todos sus maximales tienen granularidad anual y por tanto

$$granul\,(cAll) = g_{\mathbf{y}} \tag{74}$$

El sucesor n-ésimo se calcula así:

$$\mathcal{I}\,(k,g) + n \times cAll = \mathcal{I}\,(k,g) + n = \mathcal{I}\,(k+n,g)\,\forall n,\, k \in \mathbb{Z} \tag{75}$$

El conjunto universal y el fechado anual coinciden porque no se contemplan granularidades mayores que la anual. Si se considereran las granularidades correspondientes a las décadas o los siglos no sería así, pero siempre habría una granularidad máxima tal que el fechado completo correspondiente correspondería al conjunto universal.

3.3.2. Fechados básicos completos no universales: mensual, diario, horario, minutal y secundal

El resto de fechados básicos completos son los siguientes:

- Fechado mensual **cMonthly:** el conjunto de todos los IT de granularidad mensual o menos gruesa.
$$\mathcal{I}\,(k,g) \in cMonthly \Leftrightarrow g \leq g_{\mathbf{m}} \tag{76}$$
- Fechado diario **cDaily:** el conjunto de todos los IT de granularidad diaria o menos gruesa.
$$\mathcal{I}\,(k,g) \in cDaily \Leftrightarrow g \leq g_{\mathbf{d}} \tag{77}$$
- Fechado horario **cHourly:** el conjunto de todos los IT de granularidad horaria o menos gruesa.
$$\mathcal{I}\,(k,g) \in cHourly \Leftrightarrow g \leq g_{\mathbf{h}} \tag{78}$$
- Fechado minutal **cMinutely:** el conjunto de todos los IT de granularidad minutal o menos gruesa.
$$\mathcal{I}\,(k,g) \in cMinutely \Leftrightarrow g \leq g_{\mathbf{i}} \tag{79}$$
- Fechado secundal **cSecondly:** el conjunto de todos los IT de granularidad secundal o menos gruesa
$$\mathcal{I}\,(k,g) \in cSecondly \Leftrightarrow g = g_{\mathbf{s}} \tag{80}$$

Operaciones de sucesión de IT's A efectos puramente formales, no de implementación, denominamos de forma genérica $FBC(g)$ al fechado básico completo de granularidad g. A continuación se describe un método de cálculo de las funciones de sucesión para el caso $r = \mathbf{m} = 2$, es decir, $FBC(g_{\mathbf{m}}) = cMonthly$ de forma que sirva de ejemplo y de aclaración, y después se extrapola al caso general.

Caso particular: $cMonthly$ El sucesor n-ésimo en cMonthly para IT's de granularidad $h \leq g_{\mathbf{m}}$ se calcula así:

$$\mathcal{I}(k,h) + n \times cMonthly = \mathcal{I}(k+n,h)\ \forall n,\ k \in \mathbb{Z} \tag{81}$$

Para CT's de granularidad anual se tienen las siguientes reglas de cálculo:

$$\begin{array}{c} IT(y) + n \times cMonthly = (IT(y) + cMonthly) + (n-1) \times cMonthly\, \forall n \in \mathbb{Z}^{+} \\ IT(y) - n \times cMonthly = (IT(y) - cMonthly) - (n-1) \times cMonthly\, \forall n \in \mathbb{Z}^{+} \\ IT(y) + 0 \times cMonthly = I(?) \\ IT(y) + cMonthly = IT(y+1,1) = \\ IT(y) - cMonthly = IT(y-1,12) \end{array} \tag{82}$$

Si $\mathcal{I}(k, g_{\mathbf{y}}) = IT(y)$, según se puede ver en la ecuación 40 en la página 9, la expresión $\mathcal{I}\left(g_{\mathbf{m}}^{-1}(k \bullet g_{\mathbf{y}}), g_{\mathbf{m}}\right)$ es el primer mes del año y, y por tanto $\mathcal{I}\left(g_{\mathbf{m}}^{-1}((k+1) \bullet g_{\mathbf{y}}) - 1, g_{\mathbf{m}}\right)$ es el último mes del año y.

Caso general: $FBC(g)$ El sucesor n-ésimo en $FBC(g)$ para IT's de granularidad $h \leq g$ se calcula así:

$$\mathcal{I}(k,h) + n \times FBC(g) = \mathcal{I}(k+n,h)\ \forall n,\ k \in \mathbb{Z} \tag{83}$$

Para IT's de granularidad $h > g$ se calcula con las siguientes reglas:

$$\begin{array}{c} \mathcal{I}(k,h) + n \times FBC(g) = (\mathcal{I}(k,h) + FBC(g)) + (n-1) \times FBC(g)\ \forall n \in \mathbb{Z}^{+} \\ \mathcal{I}(k,h) - n \times FBC(g) = (\mathcal{I}(k,h) - FBC(g)) - (n-1) \times FBC(g)\ \forall n \in \mathbb{Z}^{+} \\ \mathcal{I}(k,h) + 0 \times FBC(g) = I(?) \\ \mathcal{I}(k,h) + FBC(g) = \mathcal{I}\left(g^{-1}((k+1) \bullet h), g\right) \\ \mathcal{I}(k,g) - FBC(g) = \mathcal{I}\left(g^{-1}(k \bullet h) - 1, g\right) \end{array} \tag{84}$$

3.3.3. Conjunto vacío: cEmpty

Llamaremos cEmpty al conjunto temporal vacío, es decir, al que no contiene ningún IT propio.

$$it \notin cEmpty\ \forall it \in \mathfrak{C}_6 \tag{85}$$

Evidentemente no se trata de un fechado aunque tampoco se puede decir que esté acotado. Se introduce en este punto preciso porque se trata del conjunto límite de la sucesión de los fechados básicos completos cuando la granularidad tiende al continuo, y por tanto se puede decir que

$$granul(cEmpty) = g_{\infty} \tag{86}$$

El sucesor n-ésimo se calcula así:

$$\mathcal{I}(k,g) + n \times cEmpty = \begin{cases} \mathcal{I}(-\infty) & si\ n < 0 \\ \mathcal{I}(?) & si\ n = 0 \\ \mathcal{I}(\infty) & si\ n > 0 \end{cases}\ \forall n,\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{87}$$

3.3.4. Fechado básico pascual: cEaster

Llamaremos cEaster al conjunto de todos IT pertenecientes a los domingos de Pascua de cualquier año, que por estar incluidos en un día del año han de tener granularidades diaria, horaria, minutal o secundal y todos sus maximales tienen granularidad diaria, luego

$$granul\left(cEaster\right) = g_{\mathbf{d}} \tag{88}$$

La definición del domingo de Pascua es relativamente sencilla: se trata del primer domingo posterior a la primera luna llena tras el equinoccio de primavera. Sin embargo los métodos con los que la iglesia calcula las efemérides astronómicas es muy primitivo y recurre a ciertas modificaciones *ad-hoc,* por lo que no siempre coincide con la definición exacta. Aunque como se ha dicho en esta implementación se utiliza la librería *libtai* se incluye aquí el siguiente código C++ que calcula de forma sencilla el domingo de Pascua para cualquier año para que el lector observe el grado de complejidad del cálculo.

```
void CalcEaster(int year, int& monthEaster, int& dayEaster)
{
  long y, g, c, x, z, d, e, n;
  g=y%19+1; // Calcular el numero aureo
  c=y/100+1;
  x=3*c/4-12;
  z=(8*c+5)/25-5;
  d=5*y/4-x-10;
  e=(11*g+20+z-x)%30; // Calcular la epacta
  if (e==25&&g>11) e--;
  // Calcular la fecha de la luna
  // segun el ciclo de Meton
  n=44-e;
  if (n<21) n+=30; // Si la fecha es anterior al 21 de
                   // marzo aumentar a abril
  n=n+7-(d+n)%7;
  if(n>31) { monthEaster = 4; dayEaster = n-31; }
  else     { monthEaster = 3; dayEaster = n; }
}
```

Merced a esta función se puede determinar el IT $easter\left(year\right) = IT(year, monthEaster, dayEaster)$ para cualquier año:

$$\mathcal{I}\left(k,g\right) \in cEaster \Leftrightarrow \mathcal{I}\left(k,g\right) \subset easter\left(year\left(k \bullet g\right)\right) \tag{89}$$

El sucesor n-ésimo se calcula así para $g \geq g_{\mathbf{d}}$:

$$\mathcal{I}\left(k,g_{\mathbf{d}}\right) + n \times cEaster = \begin{cases} easter\left(n+y\right) & si\, \mathcal{I}\left(k,g_{\mathbf{d}}\right) < easter\left(y\right) \\ easter\left(1+n+y\right) & si\, \mathcal{I}\left(k,g_{\mathbf{d}}\right) \geq easter\left(y\right) \end{cases} \wedge y = year\left(k \bullet g_{\mathbf{d}}\right) \tag{90}$$

$$\mathcal{I}\left(k,g\right) + n \times cEaster = \mathcal{I}\left(g_{\mathbf{d}}^{-1}\left(\left(k+1\right) \bullet g\right) - 1, g\right) + n \times cEaster\ \forall n,\, k \in \mathbb{Z},\, g > g_{\mathbf{d}} \tag{91}$$

Para $g < g_{\mathbf{d}}$ el sucesor se calcula siguiendo el método dado en las ecuaciones 63 y 64 en la página 13.

3.4. Funciones básicas de CT's

Los *conjuntos temporales básicos* son los creados mediante funciones que devuelven conjuntos temporales a partir de un parámetro dado, a las que llamaremos *funciones básicas de CT's.* Al igual que ocurría con las constantes, la condición de incluir los interiores de todos sus elementos se desprende de las propias definiciones de forma trivial por lo que no se demuestra en cada caso. Todos estos conjuntos temporales básicos son fechados excepto cYear.

3.4.1. Función cYear(Y)

La función cYear devuelve el conjunto temporal de todos los IT incluidos en un año dado

$$\mathcal{I}(k,g) \in cYear(Y) \Leftrightarrow year(k \bullet g) = Y \quad \forall Y \in RG_{\mathbf{y}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{92}$$

Evidentemente no es un fechado y su único maximal es $IT(Y)$ de granularidad anual y por tanto

$$granul(cYear(Y)) = g_{\mathbf{y}} \tag{93}$$

Si $\mathcal{I}(k+n,g)$ queda dentro del año Y, entonces será el sucesor n-ésimo de $\mathcal{I}(k,g)$ en $cYear(Y)$. El sucesor en $cYear(Y)$ del último instante de ese año para cada granularidad será $\mathcal{I}(\infty)$ y análogamente el predecesor del primero será $\mathcal{I}(-\infty)$. Todo esto expresado formalmente queda así:

$$\mathcal{I}(k,g)+n\times cYear(Y) = \begin{cases} \mathcal{I}(k+n,g) & si & year((k+n)\bullet g) = Y & \\ \mathcal{I}(?) & si & year(k\bullet g) \neq Y & \wedge n=0 \\ \mathcal{I}(-\infty) & si & year((k+n)\bullet g) < Y & \wedge n<0 \\ \mathcal{I}(\infty) & si & year((k+n)\bullet g) > y & \wedge n>0 \end{cases} \forall n,\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{94}$$

3.4.2. Fechados básicos parciales: cMonth(M), cDay(D), cHour(H), cMinute(Mi), cSecond(S), sWeekDay(W)

En esta sección se tratan los llamados fechados básicos parciales, en el sentido de que no recubren la recta real, y que se definen por cada valor particular que puede tomar la coordenada gregoriana correspondiente a cada granularidad desde la mensual a la secundal; añadiéndose la granularidad semanal, que no es básica propiamente hablando, pero que por su gran utilidad se tratará como si lo fuera a estos efectos.

- La función **cMonth** devuelve el conjunto temporal de todos los IT incluidos en un mes del año dado para cualquier año pasado, presente o futuro

$$\mathcal{I}(k,g) \in cMonth(M) \Leftrightarrow g \leq g_{\mathbf{m}} \wedge month(k\bullet g) = M \quad \forall M \in RG_{\mathbf{m}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{95}$$

- La función **cDay** devuelve el conjunto temporal de todos los IT incluidos en un día de mes dado para cualquier mes y cualquier año pasado, presente o futuro

$$\mathcal{I}(k,g) \in cDay(D) \Leftrightarrow g \leq g_{\mathbf{d}} \wedge day(k\bullet g) = D \quad \forall D \in RG_{\mathbf{d}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{96}$$

- La función **cHour** devuelve el conjunto temporal de todos los IT incluidos en una hora dada para cualquier día pasado, presente o futuro

$$\mathcal{I}(k,g) \in cHour(H) \Leftrightarrow g \leq g_{\mathbf{h}} \wedge hour(k\bullet g) = H \quad \forall H \in RG_{\mathbf{h}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{97}$$

- La función **cMinute** devuelve el conjunto temporal de todos los IT incluidos en una hora dada para cualquier hora de cualquier día pasado, presente o futuro

$$\mathcal{I}(k,g) \in cMinute(Mi) \Leftrightarrow g \leq g_{\mathbf{i}} \wedge minute(k\bullet g) = Mi \quad \forall Mi \in RG_{\mathbf{i}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{98}$$

- La función **cSecond** devuelve el conjunto temporal de todos los IT incluidos en una hora dada para cualquier hora de cualquier día pasado, presente o futuro

$$\mathcal{I}(k,g) \in cSecond(S) \Leftrightarrow g \leq g_{\mathbf{s}} \wedge second(k\bullet g) = S \quad \forall S \in RG_{\mathbf{s}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{99}$$

- La función **cWeekDay** devuelve el conjunto temporal de todos los IT incluidos en una día de la semana dado para cualquier semana pasada, presente o futura

$$\mathcal{I}(k,g) \in cWeekDay(W) \Leftrightarrow g \leq g_{\mathbf{d}} \wedge weekDay(k\bullet g) = W \quad \forall W \in RG_{\mathbf{w}},\ k \in \mathbb{Z},\ g \in \mathcal{G}_6 \tag{100}$$

Operaciones de sucesión de IT's A efectos puramente formales, no de implementación, denominamos de forma genérica $FBP\left(g_r,\ c_r\right)$ al fechado básico parcial de granularidad g_r y coordenada gregoriana c_r. A continuación se describe un método de cálculo de las funciones de sucesión para el caso $r = \mathbf{m} = 2$, es decir, $FBP\left(g_{\mathbf{m}},\ M\right) = cMonth\left(M\right)$ de forma que sirva de ejemplo y de aclaración, y después se extrapola al caso general.

Caso particular: $cMonth\left(M\right)$ Para IT's de granularidad mensual el sucesor y el predecesor simples en $cMonth\left(M\right)$ se calculan de forma muy sencilla

$$\begin{array}{c} IT\left(y,m\right) + cMonth\left(M\right) = \left\{ \begin{array}{cc} IT\left(y,M\right) & \forall m < M \\ IT\left(y+1,M\right) & \forall m \geq M \end{array} \right. \\ IT\left(y,m\right) - cMonth\left(M\right) = \left\{ \begin{array}{cc} IT\left(y,M\right) & \forall m > M \\ IT\left(y-1,M\right) & \forall m \leq M \end{array} \right. \end{array} = \qquad (101)$$

Para CT's de granularidad anual se tiene

$$\begin{array}{c} IT\left(y\right) + cMonth\left(M\right) = IT\left(y,12\right) + cMonth\left(M\right) \\ IT\left(y\right) - cMonth\left(M\right) = IT\left(y,1\right) - cMonth\left(M\right) \end{array} = \qquad (102)$$

Recuérdese que para extrapolar los resultados a otras granularidades es más conveniente expresar $IT\left(y,m\right) = \mathcal{I}\left(k, g_{\mathbf{m}}\right)$ y

$$\begin{array}{c} IT\left(y,1\right) = \mathcal{I}\left(g_{\mathbf{m}}^{-1}\left(k \bullet g_{\mathbf{y}}\right), g_{\mathbf{m}}\right) \\ IT\left(y,12\right) = \mathcal{I}\left(g_{\mathbf{m}}^{-1}\left(\left(k+1\right) \bullet g_{\mathbf{y}}\right) - 1, g_{\mathbf{m}}\right) \end{array} = \qquad (103)$$

Una vez internado en el propio $cMonth\left(M\right)$ el sucesor n-ésimo es trivial

$$IT\left(y,M\right) + n \times cMonth\left(M\right) = IT\left(y+n,M\right) \qquad (104)$$

De esta forma, se dispone de un método muy rápido para el cálculo del sucesor n-ésimo para las granularidades mensual y anual, o sea, para $h \geq g_{\mathbf{m}}$:

$$\begin{array}{c} \mathcal{I}\left(k,h\right) + n \times cMonth\left(M\right) = \left(\mathcal{I}\left(k,h\right) + cMonth\left(M\right)\right) + \left(n-1\right) \times cMonth\left(M\right) \\ \mathcal{I}\left(k,h\right) - n \times cMonth\left(M\right) = \left(\mathcal{I}\left(k,h\right) - cMonth\left(M\right)\right) - \left(n-1\right) \times cMonth\left(M\right) \end{array} \ \forall n \in \mathbb{Z}^{+} \qquad (105)$$

ya que tanto $\mathcal{I}\left(k,h\right) + cMonth\left(M\right)$ como $\mathcal{I}\left(k,h\right) - cMonth\left(M\right)$ pertenecen siempre a $cMonth\left(M\right)$ para $h \geq g_{\mathbf{m}}$.

Caso general: $FBP\left(g_r,\ c_r\right)$ Para IT's de granularidad g_r el sucesor y el predecesor simples en $FBP\left(g_r,\ c_r\right)$ se calculan de forma muy sencilla

$$\begin{array}{c} IT\left(c_1',\ ...,\ c_{r-1}',\ c_r'\right) + FBP\left(g_r,\ c_r\right) = \left\{ \begin{array}{cc} IT\left(c_1',\ ...,\ c_{r-1}',\ c_r', c_r'\right) & \forall c_r' < c_r \\ IT\left(c_1',\ ...,\ c_{r-1}'+1,\ c_r'\right) & \forall c_r' \geq c_r \end{array} \right. \\ IT\left(c_1',\ ...,\ c_{r-1}',\ c_r'\right) - FBP\left(g_r,\ c_r\right) = \left\{ \begin{array}{cc} IT\left(c_1',\ ...,\ c_{r-1}',\ c_r', c_r'\right) & \forall c_r' > c_r \\ IT\left(c_1',\ ...,\ c_{r-1}'-1,\ c_r'\right) & \forall c_r' \leq c_r \end{array} \right. \end{array} = \qquad (106)$$

Para CT's de granularidad $h > g_r$ se tiene

$$\begin{array}{c} \mathcal{I}\left(k,h\right) + FBP\left(g_r,\ c_r\right) = \mathcal{I}\left(g_r^{-1}\left(\left(k+1\right) \bullet h\right) - 1, g_r\right) + FBP\left(g_r,\ c_r\right) \\ \mathcal{I}\left(k,h\right) - FBP\left(g_r,\ c_r\right) = \mathcal{I}\left(g_r^{-1}\left(k \bullet h\right), g_r\right) - FBP\left(g_r,\ c_r\right) \end{array} = \qquad (107)$$

Si tenemos $\mathcal{I}\left(k, g_r\right) \in FBP\left(g_r,\ c_r\right)$ el sucesor n-ésimo se calcula simplemente desplazándolo su punto de inicio $n-1$ veces en la granularidad g_{r-1}, que es la inmediatamente más gruesa que g_r, y una vez en el propio $FBP\left(g_r,\ c_r\right)$, lo cual se puede expresar como

$$\begin{array}{l} \mathcal{I}\left(k, g_r\right) + n \times FBP\left(g_r,\ c_r\right) = \mathcal{I}\left(l+n-1, g_{r-1}\right) + FBP\left(g_r,\ c_r\right) \\ \mathcal{I}\left(k, g_r\right) - n \times FBP\left(g_r,\ c_r\right) = \mathcal{I}\left(l-n+1, g_{r-1}\right) - FBP\left(g_r,\ c_r\right) \end{array} \quad \forall n \in \mathbb{Z}^{+} \tag{108}$$

donde $\mathcal{I}\left(l, g_{r-1}\right)$ es el contenedor de $\mathcal{I}\left(k, g_r\right)$ en la granularidad g_{r-1}, tal y como se describe en la ecuación 39 en la página 9.

De esta forma, se dispone de un método muy rápido para el cálculo del sucesor n-ésimo para las granularidades $h \geq g_r$:

$$\begin{array}{l} \mathcal{I}\left(k, h\right) + n \times FBP\left(g_r,\ c_r\right) = \left(\mathcal{I}\left(k, h\right) + FBP\left(g_r,\ c_r\right)\right) + (n-1) \times FBP\left(g_r,\ c_r\right) \\ \mathcal{I}\left(k, h\right) - n \times FBP\left(g_r,\ c_r\right) = \left(\mathcal{I}\left(k, h\right) - FBP\left(g_r,\ c_r\right)\right) - (n-1) \times FBP\left(g_r,\ c_r\right) \end{array} \tag{109}$$

Para $h < g_r$ el sucesor se calcula siguiendo el método dado en las ecuaciones 63 y 64 en la página 13.

3.5. Operaciones Booleanas

Las siguientes operaciones entre conjuntos temporales le dan a $\mathcal{P}\left(\mathfrak{C}_6\right)$ una estructura de álgebra de Boole. Con ellas y otras más que se verán más adelante se pueden construir expresiones que den forma a las secuencias temporales más complicadas que se pueden encontrar en el análisis de series temporales.

3.5.1. Operación Suma de CT's

La *suma maximal* o simplemente suma de dos conjuntos temporales es el conjunto generado por la sucesión resultante de la unión disjunta de las sucesiones maximales de ambos

$$Ct_1 + Ct_2 = \underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}} \tag{110}$$

Dicho de otra forma: el maximal de la suma de CT's es la suma disjunta de sus maximales.

$$\overbrace{Ct_1 + Ct_2} = \overbrace{Ct_1} \oplus \overbrace{Ct_2} \tag{111}$$

Es decir, los maximales de $Ct_1 + Ct_2$ son los maximales de Ct_1 que no están incluidos en ningún maximal de Ct_2 y los maximales de Ct_2 que no están incluidos en ningún maximal de Ct_1. Por lo tanto sólo se puede asegurar en términos generales que

$$granul\left(Ct_1 + Ct_2\right) \geq \text{mín}\left\{granul\left(Ct_1\right),\ granul\left(Ct_2\right)\right\} \tag{112}$$

Proposición: La suma de conjuntos temporales coincide con la unión de ambos conjuntos

$$Ct_1 + Ct_2 = Ct_1 \cup Ct_2 \tag{113}$$

Demostración: Los elementos de $\underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}}$ están incluidos en elementos que pertenecen a Ct_1 o a Ct_2 por lo que también pertenecen a uno u otro, y se tiene que $\underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}} \subset Ct_1 \cup Ct_2$. Si $it \in Ct_1$ entonces $\exists it_1 \in \overbrace{Ct_1} \subset \underbrace{\overbrace{Ct_1}} \subset \underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}} \perp it \subset it_1$ luego $it \in \underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}}$. Del mismo modo se demuestra que si $it \in Ct_2$ entonces $it \in \underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}}$. Luego $Ct_1 \cup Ct_2 \subset \underbrace{\overbrace{Ct_1} \oplus \overbrace{Ct_2}}$.

Obviamente , la suma de fechados es también un fechado.

El sucesor de un IT en la suma de dos CT's es el mínimo de los sucesores en cada uno y el predecesor es el máximo.

3.5.2. Operación Producto de CT's

El *producto maximal* o simplemente producto de dos conjuntos temporales es el conjunto generado por la intersección de las sucesiones maximales de ambos

$$Ct_1 * Ct_2 = \underbrace{\overbrace{Ct_1} \otimes \overbrace{Ct_2}} \tag{114}$$

Dicho de otra forma: el maximal del producto de CT's es el producto disjunto de sus maximales.

$$\overbrace{Ct_1 * Ct_2} = \overbrace{Ct_1} \otimes \overbrace{Ct_2} \tag{115}$$

Es decir, los maximales de $Ct_1 * Ct_2$ son los maximales comunes a Ct_1 y Ct_2 y se puede asegurar que

$$granul\,(Ct_1 * Ct_2) = \text{mín}\,\{granul\,(Ct_1)\,,\ granul\,(Ct_2)\} \tag{116}$$

PROPOSICIÓN: El producto de dos conjuntos temporales coincide con la intersección de ambos conjuntos

$$Ct_1 * Ct_2 = Ct_1 \cap Ct_2 \tag{117}$$

DEMOSTRACIÓN: Los elementos de $\underbrace{\overbrace{Ct_1} \otimes \overbrace{Ct_2}}$ están incluidos en elementos que pertenecen a Ct_1 y a Ct_2 por lo que también pertenecen a uno y a otro, y se tiene que $\underbrace{\overbrace{Ct_1} \otimes \overbrace{Ct_2}} \subset Ct_1 \cap Ct_2$. Si $it \in Ct_1 \cap Ct_2$ entonces existe un it' que es elemento maximal de $Ct_1 \cap Ct_2$, o sea $\exists it' \in \overbrace{Ct_1 \cap Ct_2} \perp it \subset it'$. Luego it' pertenece a ambos conjuntos y no hay ningún otro elemento en ellos que lo contenga, por lo tanto es maximal en cada uno de ellos, es decir, $it' \in \overbrace{Ct_1} \cap \overbrace{Ct_2} = \overbrace{Ct_1} \otimes \overbrace{Ct_2}$. Por lo tanto $it \in \underbrace{\overbrace{Ct_1} \otimes \overbrace{Ct_2}}$ y $Ct_1 \cap Ct_2 \subset \underbrace{\overbrace{Ct_1} \otimes \overbrace{Ct_2}}$.

La granularidad del producto de CT's es obviamente la mínima de las granularidades de cada uno

$$granul\,(Ct_1 * Ct_2) = \text{mín}\,\{granul\,(Ct_1)\,,\ granul\,(Ct_2)\} \tag{118}$$

El producto de fechados puede ser un fechado o no serlo.

El sucesor de un IT en el producto de dos CT's es algo más complicado que el de la suma . Primero se calcula el sucesor en $it' = it + Ct_1$. Si $it' \in Ct_2$ ya está hecho. Si no, se calcula $it'' = it' + Ct_2$. Si $it'' \in Ct_1$ ya está hecho y si no se pone $it \leftarrow it''$ y se continua calculando sucesores alternativamente en uno y otro conjunto hasta que se encuentre uno que pertenezca al otro conjunto o nos devuelva un IT impropio. De forma análoga se calcula el predecesor retrocediendo en cada uno de los CT's y haciendo las mismas comprobaciones en cada iteración.

Este caracter recursivo da lugar a un ciclo infinito cuando el resultado de una intersección es vacío. Por ello se debe disponer de un mecanismo que analize simbólicamente las expresiones para evitar esa situación. En cualquier caso sería conveniente dar un mensaje de advertencia cuando se sobrepasara determinado número de iteraciones considerado como excesivo. También sería bueno para el ciclo y dar un mensaje de error a partir de un úmero máximo de iteraciones. Ambos límites deberían ser definibles por el usuario.

3.5.3. Operación Diferencia de CT's

La *diferencia maximal*, o simplemente diferencia de dos conjuntos temporales es el conjunto temporal generado por la diferencia disjunta de las sucesiones maximales de ambos conjuntos

$$Ct_1 \ominus Ct_2 = \underbrace{\overbrace{Ct_1} \ominus \overbrace{Ct_2}} \tag{119}$$

Dicho de otra forma: el maximal de la diferencia de CT's es la diferencia disjunta de sus maximales.

$$\overbrace{Ct_1 \ominus Ct_2} = \overbrace{Ct_1} \ominus \overbrace{Ct_2} \tag{120}$$

Es decir, los maximales de $Ct_1 \ominus Ct_2$ son los maximales de Ct_1 que no incluyen a los de Ct_2. Por lo tanto sólo se puede asegurar que

$$granul\,(Ct_1 \ominus Ct_2) \geq granul\,(Ct_1) \tag{121}$$

Se demuestra de forma trivial que la diferencia maximal no coincide con la diferencia usual de conjuntos

$$Ct_1 \ominus Ct_2 \neq Ct_1 - Ct_2 = \{it \in \mathfrak{C}_6 | it \in Ct_1 \wedge it \notin Ct_2\} \tag{122}$$

Por ejemplo $cDaily - cHourly$ contendría el IT y2005m1d1 pero no el y2005m1d1h1 que pertenece al interior del primero. Luego $cDaily - cHourly$ no es un conjunto temporal mientras que $cDaily \ominus cHourly$ sería el conjunto vacío pues cada IT maximal de $cDaily$ contiene IT's maximales de $cHourly$ y serían todos eliminados.

Otro ejemplo: el conjunto $cDaily \ominus cHour\,(8) = cHourly \ominus cHour\,(8)$ contendría todas las horas excepto la octava de cada día.

La diferencia de fechados puede ser un fechado o no serlo.

Para calcular el sucesor de un IT en la diferencia de fechados primero se calcula el sucesor en $it' = it + Ct_1$. Si $it' \notin Ct_2$ ya está hecho. Si no, se calcula $it'' = it' + Ct_1$. Si $it'' \notin Ct_2$ ya está hecho y si no, se continua calculando sucesores en Ct_1 hasta que se encuentre uno que no esté en Ct_2 o nos devuelva un IT impropio. De forma análoga se calcula el predecesor retrocediendo en Ct_1 y haciendo las mismas comprobaciones en cada iteración. Considérese válido aquí todo lo referente a los ciclos infinitos explicado en el apartado anterior.

3.6. Operadores de Traslación en un Fechado

Los operadores de traslación de conjuntos construyen un CT basándose en desplazar todos sus IT's a lo largo de un fechado.

3.6.1. Operación cSucc

Se define el conjunto sucesor n-ésimo de un conjunto Ct en un fechado F, y se expresa $cSuccInCT\,(Ct, n, F)$ ó $Ct + n \times F$, como el conjunto temporal generado por la sucesión disjunta obtenida al aplicar el sucesor n-ésimo en el fechado F a todos los IT maximales de Ct, al que llamaremos conjunto de origen de la traslación.

$$\overbrace{Ct + n \times F} = \left\{ it \in F \mid \exists\, it' \in \overbrace{Ct} \wedge it = it' + n \times F \right\} \tag{123}$$

La granularidad del conjunto sucesor n-ésimo es la mínima de las granularidades de Ct y F.

$$granul\,(Ct + n \times F) = \text{mín}\,\{granul\,(Ct)\,,\, granul\,(F)\} \tag{124}$$

El conjunto sucesor de un fechado en un fechado es también un fechado.

Cálculo de la función de pertenencia a $Ct + n \times F$ El método de cálculo será constructivo, es decir, se intentará encontrar el IT de Ct cuyo sucesor n-ésimo es el IT dado. Si se consigue se devuelve *cierto* y si no se consigue se devuelve *falso*.

Sea $it \in F$ y supongamos que $n \in \mathbb{Z}^+$, entonces llamamos

$it' = it - (n-1) \times F$	al predecesor $(n-1)$-ésimo en F de it
$it'' = it' - Ct \in Ct$	al predecesor en Ct de it'
$it''' = it - n \times F = it' - F \in Ct$	al predecesor n-ésimo en F de it

Entonces se se puede asegurar que

$$it \in Ct + n \times F \Leftrightarrow it \in F \wedge it''' \leq it'' \tag{125}$$

Para comprobarlo sólo hay que visualizar la secuencia temporal que forman los IT's definidos

$$it''' = it - n \times F \leq it'' < it' = it - (n-1) \times F < \cdots < it - F < it \tag{126}$$

Salta a la vista que existe $it'' \in Ct$ tal que $it'' + n \times F = it$.

Para $Ct - n \times F$ con $n \in \mathbb{Z}^+$se puede desarrollar un argumento análogo al anterior definiendo $it' = it + (n-1) \times F$ y $it'' = it' + Ct \in Ct$ para formar la secuencia

$$it' + F = it + n \times F > it'' \leq it' = it + (n-1) \times F > \cdots > it + F > it \tag{127}$$

Cuando $n = 0$ se tiene $Ct \pm 0 \times F = Ct * F$ y, en resumen, se puede asegurar que $\forall\, n \in \mathbb{Z}$ se cumple que

$$it \in Ct + n \times F \Leftrightarrow it \in F \wedge \begin{cases} n = 0 & \wedge & it \in Ct & & & \vee \\ n \neq 0 & \wedge & it' = it - n \times F \in Ct & & & \vee \\ n > 0 & \wedge & it' \notin Ct & \wedge & it' - Ct > it' - F & \vee \\ n < 0 & \wedge & it' \notin Ct & \wedge & it' + Ct < it' + F & \end{cases} \tag{128}$$

Cada llamada a la función de pertenencia a $S = Ct + n \times F$ requiere un coste computacional que se puede expresar así.

$$C(\in S) \leq C(\in F) + C(\in Ct) + C(\pm Ct) + C(\pm n \times F) \tag{129}$$

Cálculo del sucesor h-ésimo de un IT en $Ct+n\times F$ En el apartado anterior se ha calculado el $it'' \in Ct$ tal que $it'' + n \times F = it$ dado un $it \in F$. Si $it \in F$

Pues bien, si calculamos $\dot{it} = it'' + Ct$, es evidente que $\dot{it} + n \times F \in Ct + n \times F$ y que $\dot{it} + n \times F > it$ así como que no puede haber ningún otro elemento de $Ct + n \times F$ situado entre $it = it'' + n \times F$ y $\dot{it} + n \times F$ que ha de ser forzosamente el sucesor de it en $Ct + n \times F$.

Por lo tanto, para calcular el sucesor o el predecesor simple de un IT se tendrá en cuenta lo siguiente

$$\begin{array}{c} it + (Ct + n \times F) = (it'' + Ct) + n \times F \\ it + (Ct + n \times F) = (it'' - Ct) + n \times F \end{array} \tag{130}$$

El coste computacional en este caso será

$$C(\pm S) \leq C(\in F) + C(\in Ct) + 2 \cdot C(\pm Ct) + C(\pm 2 \cdot n \times F) \tag{131}$$

Una vez calculado el primer sucesor o predecesor, los siguientes son simplemente sucesores o predecesores en en el fechado F . Así pues, se tienen las siguientes fórmulas que reducen el cálculo del sucesor h-ésimo de un IT a una sola ejecución de la función sucesor seguida por el sucesor $h-1$-ésimo en el fechado F . Lo mismo ocurre si se quiere el predecesor h-ésimo

$$\begin{array}{c} it + h \times (Ct + n \times F) = (it + (Ct + n \times F)) + (h-1) \times F\ \forall h \in \mathbb{Z}^+ \\ it - h \times (Ct + n \times F) = (it - (Ct + n \times F)) - (h-1) \times F\ \forall h \in \mathbb{Z}^+ \end{array} \tag{132}$$

El coste computacional total será

$$C(\pm h \times S) \leq C(\in F) + C(\in Ct) + 2 \cdot C(\pm Ct) + C(\pm (2 \cdot n + h - 1) \times F) \tag{133}$$

3.6.2. Operación cRange

La operación rango de sucesores o cRange se basa en la anterior cSucc y se define como la unión de sucesores consecutivos en un rango de números enteros

$$\begin{array}{l} Ct+[n,m]\times F = cRangeInCT(Ct,\, n,\, m,\, F) \;=\; \cup_{j=n}^{m}\left(Ct+j\times F\right)\; \forall n\leq\, m\in\mathbb{Z} \\ Ct+[n,m]\times F \;=\; \{it\in F|\exists j=n\ldots m, it'\in Ct\wedge it=it'+j\times F\} \end{array} \tag{134}$$

La granularidad del conjunto cRange es la mínima de las granularidades de Ct y F.

$$granul\left(cRangeInCT\left(Ct,n,F\right)\right)=\text{mín}\left\{granul\left(Ct\right),\, granul\left(F\right)\right\} \tag{135}$$

Para no repetir los cálculos es conveniente implementar los métodos de pertenencia y sucesión de forma explícita en lugar de recurrir a la suma de los cSucc. De esta forma se reaprovechan los cálculos de los sucesores consecutivos de forma recursiva

$$it'_j = it\pm j\times F = it'_{j-1}\pm F \tag{136}$$

Como los $Ct+n\times F$ se calculan de forma diferente según sea n positiva, negativa o nula puede ser necesario separar $Ct+[n,m]\times F$ de forma que se tengan bloques de rangos $Ct+[n',m']\times F$ todos positivos o todos negativos.
El coste operacional de evaluar $it\in RS=Ct+[n,m]\times F$ será

$$C\left(\in RS\right)\leq C\left(\in F\right)+L\cdot C\left(\in Ct\right)+C\left(\pm Ct\right)+C\left(\pm\left(M+1\right)\times F\right) \tag{137}$$

donde $L=m-n+1$ es la longitud del rango y $M=\{|m|+|n|\;\; si\; m\cdot n\leq 0, |m-n|\;\; si\; m\cdot n=0\}$
El coste operacional de evaluar $it+RS$ será

$$C\left(\pm RS\right)\leq C\left(\in F\right)+\left(L+1\right)\cdot C\left(\in Ct\right)+C\left(\pm Ct\right)+C\left(\pm\left(2\cdot M+1\right)\times F\right) \tag{138}$$

El coste operacional de evaluar $it+h\times RS$ será

$$Cost\left(\times RS\right)\leq Cost\left(\in F\right)+\left(L+1\right)\cdot Cost\left(\in Ct\right)+Cost\left(\pm Ct\right)+Cost\left(\pm\left(2\cdot M+h\right)\times F\right) \tag{139}$$

3.6.3. Operación cPeriodic

La operación cPeriodic devuelve todos los IT cuya distancia en un fechado F a un $\mathcal{I}\left(k,g\right)$ fijo sea múltiplo de una periodicidad p.

$$cPeriodicSucc(\mathcal{I}\left(k,g\right),\, p,\, F) \;=\; \{it'\in F|\exists n\in\mathbb{Z}\perp it'=\mathcal{I}\left(k,g\right)+\left(p\cdot n\right)\times F\} \tag{140}$$

Su granularidad es la mínima de las de it y F

$$cPeriodicSucc(it,\, p,\, F)=\text{mín}\left\{granul\left(it\right),\, granul\left(F\right)\right\} \tag{141}$$

3.7. Operadores de Auto-Traslación

Los operadores de auto-traslación de conjuntos construyen un CT basándose en desplazar todos sus IT's maximales en su propia granularidad.

3.7.1. Operación cSelfSucc

Se define el conjunto n-autosucesor de un conjunto Ct, al que llamaremos conjunto de origen del desplazamiento, como el conjunto temporal generado por la sucesión disjunta resultante de desplazar los IT's maximales de Ct, un total de n veces cada IT maximal en su propia granularidad.

$$cSelfSucc\,(Ct, n) = Ct + n = \{\mathcal{I}\,(k, g) \in \mathfrak{C}_6 | \mathcal{I}\,(k - n, g) \in Ct\} \tag{142}$$

La granularidad del conjunto n-desplazado es obviamente la misma que la de Ct.

$$granul\,(cSelfSucc\,(Ct, n)) = granul\,(Ct) \tag{143}$$

3.7.2. Operación cSelfRange

La operación rango de desplazamientos o cRangeMove se basa en la anterior cMove y se define como la unión de desplazamientos consecutivos en un rango de números enteros

$$\begin{aligned} cRangeMove(Ct,\, n,\, m) &= \textstyle\bigcup_{k=n}^{m} Ct + k \;\forall n,\, m \in \mathbb{Z} \\ cRangeMove(Ct,\, n,\, m) &= \{\mathcal{I}\,(k, g) \in \mathfrak{C}_6 | \exists j = n...m \bot \mathcal{I}\,(k - j, g) \in Ct\} \end{aligned} \tag{144}$$

La granularidad del conjunto cRange es evidentemente la de Ct.

$$granul\,(cMove\,(Ct, n)) = granul\,(Ct) \tag{145}$$

3.8. Operadores de Desplazamiento Granular

Los operadores de desplazamineto de conjuntos construyen un CT basándose en desplazar todos sus IT's un número de unidades en una granularidad dada.

3.8.1. Operación cSuccInG

3.8.2. Operación cRangeInC