Quantitative Time Logic

Wolfgang P. Kowalk, Universität Oldenburg

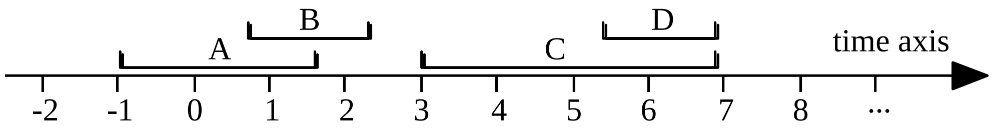
Postfach 2503, 25111 Oldenburg, Germany

Keywords: *Extensions of FDTs, real-time aspects, Consistency relations, Practical algorithms and tool support*

Abstract: *We extend Allen's time logic to a system of quantitative assertions on relationships of intervals. To do this, we use a new notation that allows simple construction and interpretation of relationship operators, introducing a simple way to compute new relationships between periods and finally extends to generalised "multiple-periods" as well as quantitative relationships between periods.*

1 Introduction

Time logic as introduced by Allen [Allen84] can be used to state relationships between intervals on the real numerical axis. Interpreting the real numerical axis as time, we can state relationships between time periods. Thus, for example, we can state that period B does not start before period A starts, or that periods C and D terminate at the same time.



Allen introduces some relationships of this type and shows that all other required relationships between intervals can be expressed by disjunctive combinations of these, e.g. either "A ends before B starts" or "B ends before A ends".

However, Allen introduces letters for representing these relationships, like e or g. This yields statements as $A\,(e, g, f)\,B$, which are not very easy to interpret. Also, these relationships can be inconsistent. It is usually not easy to check for consistency, since all combinations of relationships must be considered. If there are relationships between intervals $A_1, A_2, A_3, \ldots, A_N$, the number of tests to prove inconsistency can grow exponentially with the number N of intervals. Also, there are other severe disadvantages of Allen's method. It is for example not possible to extend this notation straightforward to quantitative statements on relationships between intervals.

To overcome these restrictions, in the next section of this paper we introduce a notation that is easier to construct than Allen's set of letters and easier to interpret.

The third section gives a simple but effective and efficient method to compute a new relationship between A and C provided relationships between A and B as well as B and C are given. This "*Aristotelian syllogism*" like conclusion has to be performed by Allen’s method by use of big tables, exhibiting for each combination of two relations its "*product*". Our technique suggests a simple multiplication method, similar to matrix-multiplication in linear algebra.

A further extension is given in section 4. It shows, how quantitative relationships between periods can be introduced. They are "*self-explaining*" in the following sense: it is easy to state and understand such a relationship. Multiplication rules for *syllogism* like conclusions can be derived from the corresponding simpler rules for non-quantitative relationships between periods.

2 Self-explaining notation for relationships between intervals

In Allen‘s time logic a relationship between two intervals on the real axis is expressed by combining some basic relationships. This will be done here in a similar way, however using more familiar symbols and more systematic notations.

Let there be intervals on the real axis. Let us denote them as A, B, C … using capital letters. For interval A we write A_1 for the first or left point, A_2 for the second or right point; thus in a standard mathematical notation we can write $A=\left[A_1,A_2\right]$. Employing numerical indices helps to extend this to multiple-intervals like $A=\left[A_1,A_2,A_3,A_4\right]$, which are straightforward extensions of the classical interval notion. Of course, necessary conditions between the real points A_1,A_2,A_3,A_4 are $A_1 \le A_2 \le A_3 \le A_4$.

Definition

Intervals are pairs of real numbers $A=\left[A_1,A_2\right]$, where $A_1 \le A_2$ and $A_1,A_2 \in \Re$.

Multiple-intervals of degree n (or n-multiple-intervals or shorter: **n-intervals)** are sequences of n real numbers $A=\left[A_1,A_2,\ldots,A_N\right]$, where $A_1 \le A_2 \le \ldots \le A_N$ and $A_1,A_2,\ldots,A_N \in \Re$.

Multiple-intervals can be used to model consecutive actions, like initial, intermediate and final periods of a machine. This technique simplifies temporal models of empirical systems.

A qualitative description of the relationship between two intervals $A=\left[A_1,A_2\right]$ and $B=\left[B_1,B_2\right]$ can be stated by expressing the relationships between the real points $\mathrm{A}_1,\mathrm{A}_2,B_1,\mathrm{B}_2$. We know by definition that $A_1 \le A_2$ and $B_1 \le B_2$. However, relationships between the other four pairs of points must be stated by the model. Using lower case letters $a,b,\ldots$ for unassigned relationships we have to specify the relationships $A_1 a B_1$, $A_1 b B_2$, $A_2 c B_1$, $A_2 d B_2$.

There are three independent relationships between points, namely „left“, „on“ and „right“. Combining them yields eight possible relationships between two points:

Definition

There are eight possible qualitative relationships between two real points X and Y. The following table states them as well as the symbols we utilise to display them.

- $X < Y$ means: Point X is left of point Y.
- $X \le Y$ means: Point X is left of point Y or on point Y.
- $X = Y$ means: Point X is on point Y.
- $X \ne Y$ means: Point X is left or right of point Y.
- $X \ge Y$ means: Point X is right of point Y or on point Y.
- $X > Y$ means: Point X is right of point Y.
- $X \cong Y$ means: Point X is right, left or on point Y, i.e. all relationships are feasible.
- $X \varnothing Y$ means: Point X is neither right, nor left nor on point Y, i.e. no relationship is feasible

Instead of "left" we can say "before", "less" or "lower"; instead of "right" we state sometimes "after", "higher" or "larger".

The last two relationships are required for some formal conclusions that will be detailed later.

Now we have the framework to define a notation of relationships between intervals. As has been seen from the previous discussion there are four unassigned relations that can be written in a matrix form,

$$A\left\langle \frac{a}{c} \middle| \frac{b}{d} \right\rangle B \quad \Leftrightarrow \quad A_1 a B_1 \wedge A_1 b B_2 \wedge A_2 c B_1 \wedge A_2 d B_2 .$$

The symbols in the first line $\langle a,b \rangle$ are used to state the relationship between point A_1 and interval B; those in the second line $\langle c,d \rangle$ express the relationship between point A_2 and B. In a similar way, the symbols in the first column relate interval A to B_1, those in the second one A to B_2. Beyond pure formal analogy, this order is required for the computational properties of the symbols that will be introduced in the next section.

To baptise them we define

Definition

Let A and B be two intervals. Let us state a relationship between A and B by the symbol

$$A\left\langle \frac{a}{c} \middle| \frac{b}{d} \right\rangle B \quad \Leftrightarrow \quad A_1 a B_1 \wedge A_1 b B_2 \wedge A_2 c B_1 \wedge A_2 d B_2 .$$

where $a,b,c,d \in \{<,\le,>,\ge,=,\ne,\cong,\varnothing\}$. Then $\left\langle \frac{a}{c} \middle| \frac{b}{d} \right\rangle$ is called a **temporator**, since it can be interpreted as a <u>tempo</u>ral relationship ope<u>rator</u>.

People interested in interval logic might call these symbols **periodators** (sind intervators or so sounds ugly). Anyway, the temporator is a straightforward extension of the usual relationship between elements in a totally ordered set. This follows from the following considerations: Besides expressing relationships between intervals, temporators can be used to express relationships between any multiple-interval, among which are also 1-intervals. Thus using a 1×1-matrix, we can write $A \le B$, where A and B both are 1-intervals. Also we can express a relationship between a 1-interval A and a 2-interval . For example, if $A > B_1$ and $A \le B_2$, then we can write $A\langle > | \le \rangle B$.

To become familiar with the newly introduced notation, we give some examples.

Examples

Let us use temporal notions, considering 2-intervals. Let A and B start at the same time. This means of course that $A_1 = B_1$, using the "start at the same time" temporator we can write

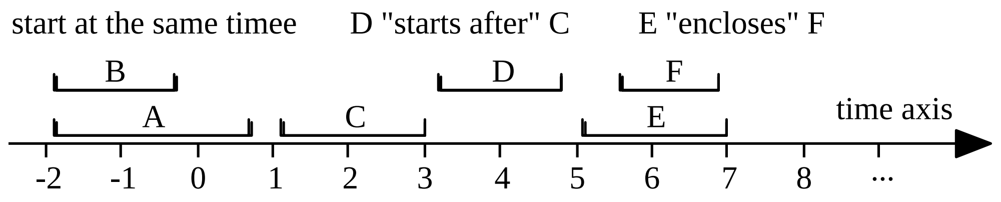
$$A\left\langle \frac{=}{\cong} \middle| \frac{\cong}{\cong} \right\rangle B .$$

Now, let D start not before C ends. This means that $C_2 \le D_1$, or using the "ends before" temporator we find

$$C\left\langle \frac{\cong}{\le} \middle| \frac{\cong}{\cong} \right\rangle D .$$

If F starts and ends during E, we have that $E_1 < F_1$ and that $E_2 > F_2$. In this case the "encloses" temporator may be used,

$$E\left\langle \frac{<}{\cong} \middle| \frac{\cong}{>} \right\rangle F .$$

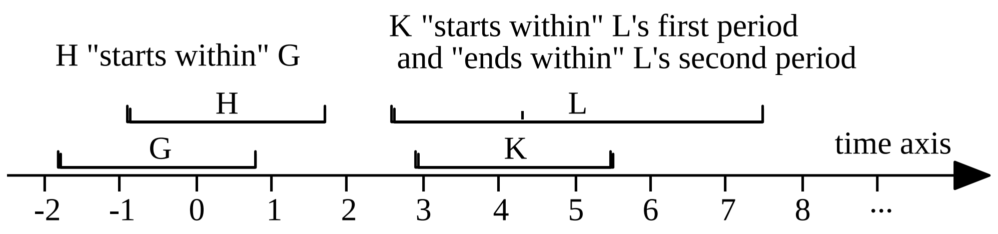


The next example requires H to start during G, but not start with G. This means that $G_1 < H_1$ and that $G_2 \ge H_1$. Here we may use the "starts" temporator,

$$G\left\langle \frac{<}{\ge} \middle| \frac{\cong}{\cong} \right\rangle H .$$

Now let K be a 2-interval and L be a 3-interval. We want K to start during the first period of L and end during the second one. This means that $L_1 < K_1 < L_2 < K_2 < L_3$, so that follows

$$K\left\langle \frac{>}{\cong} \middle| \frac{<}{>} \middle| \frac{\cong}{<} \right\rangle L .$$


H "starts within" G
K "starts within" L's first period
and "ends within" L's second period
H
L
G
K
time axis
-2
-1
0
1
2
3
4
5
6
7
8
···

The examples show, how an obvious relationship between intervals can be expressed using temporators. It also shows that expressing the corresponding meaning in plain English is usually cumbersome and inexact. Thus the temporator notation extends spoken language, which helps to clarify one‘s intentions.

In the previous examples we have give one or more relationships between intervals, e.g. $A_1 < B_1$ and $A_2 > B_2$, and derived a temporator from them. However, we know that there are more relationships than these ones, since we always have: $A_1 \le A_2$ and $B_1 \le B_2$. Thus we have indeed four relationships from which other relationships might be concluded. For the previous example we have the relationships

$$A_1 \le A_2,\ B_1 \le B_2,\ A_1 < B_1,\ A_2 > B_2,\ A_1 < B_2,\ A_2 > B_1$$

The last two inequalities follow from $A_1 < B_1$, $B_1 \le B_2$, and $A_2 > B_2$ and $B_2 \ge B_1$. Thus we can take as a temporator

$$A\left\langle \begin{array}{c|c} < & < \\ \hline > & > \end{array} \right\rangle B,$$

which has exactly the same meaning as

$$A\left\langle \begin{array}{c|c} < & \cong \\ \hline \cong & > \end{array} \right\rangle B.$$

This follows from the additional "hidden" properties of the considered relationships. To analyse this more systematically we introduce the following „temporal circles“

$$\left\langle \begin{array}{ccc} < & \Rightarrow & < \\ \Uparrow & \langle < \rangle & \Uparrow \\ < & \Rightarrow & < \end{array} \right\rangle \text{ and } \left\langle \begin{array}{ccc} > & \Leftarrow & > \\ \Downarrow & > & \Downarrow \\ > & \Leftarrow & > \end{array} \right\rangle,$$

which means that starting from the left lower field you can propagate "<" to the right upper field, or starting there you can propagate ">" to the left lower field. For example from $A\left\langle \begin{array}{c|c} \cong & \cong \\ \hline < & \cong \end{array} \right\rangle B$ follows: $A\left\langle \begin{array}{c|c} < & \cong \\ \hline < & \cong \end{array} \right\rangle B$, since $A_1 \le A_2$ and $A_2 < B_1$ implies that $A_1 < B_1$. We encourage the reader to check the other relationships; there are eight all together.

A symbol "<" can be propagated to the same symbol "<"; a symbol "$\le$" is to propagate to "$\le$" as well, since for example from $A_2 \le B_1$ and $A_1 \le A_2$ follows $A_1 \le B_1$, i.e. $A_1 = B_1$ is possible, although not necessary. To refine this, we introduce the notion of a non-zero-interval, the length of which is always larger than 0. If A is a non-zero-interval, then we get the stronger condition $A_1 < A_2$, so that from $A_2 \le B_1$ follows: $A_1 < B_1$. Here, the case $A_1 = B_1$ is excluded which makes the statement stronger.

If symbol "=" is given, it can be propagated as well, since from $A_2 = B_1$ and $A_1 \le A_2$ follows $A_1 \le B_1$. Thus "=" is propagated to "$\le$", if it is shifted up or right; it is propagated to "$\ge$", if it is shifted left or down in the temporator.

Symbol "$\ne$" cannot be propagated, since from $A_2 \ne B_1$ and $A_1 \le A_2$ follows nothing about the relationship between A_1 and B_1. Symbol "$\cong$" cannot be propagated, as well.

Finally we discuss the problem that a symbol is propagated to the same field where another symbol resides, e.g. $A\left\langle\begin{array}{c|c} > & \cong \\ \hline < & \cong \end{array}\right\rangle B$ implies by propagation $A\left\langle\begin{array}{c|c} < \wedge > & \cong \\ \hline < & \cong \end{array}\right\rangle B$. Here, both must hold, $A_1 < B_1$ as well as $A_1 > B_1$, which is impossible. Of course, this follows from the precondition, since $A\left\langle\begin{array}{c|c} > & \cong \\ \hline < & \cong \end{array}\right\rangle B$ means $A_1 > B_1$, $A_2 < B_1$, which implies $A_2 < B_1 < A_1$, contradicting $A_1 \le A_2$.

Here we would have to write: From $A\left\langle\begin{array}{c|c} > & \cong \\ \hline < & \cong \end{array}\right\rangle B$ follows by propagation $A\left\langle\begin{array}{c|c} \varnothing & \cong \\ \hline < & \cong \end{array}\right\rangle B$, which describes always an impossible situation. Thus propagation can be used to find statements on temporators, like inconsistency.

If in the latter example we would have had: $A\left\langle\begin{array}{c|c} \ge & \cong \\ \hline \le & \cong \end{array}\right\rangle B$, the conclusion by propagation would of course have been $A\left\langle\begin{array}{c|c} = & \cong \\ \hline \le & \cong \end{array}\right\rangle B$, or by propagating "=" down follows $A\left\langle\begin{array}{c|c} = & \cong \\ \hline = & \cong \end{array}\right\rangle B$, which follows also from the assumptions, as can be seen easily. Thus propagation can sometimes reveal knowledge about a relationship not given explicitly. Here, A is a zero-interval, which follows from the precondition that A starts not before the beginning of B and does not end after the beginning of B.

Definition

Given a temporator, **propagation** means to shift the symbols "<" or "$\le$" right or up, and to shift the symbols ">" or "$\ge$" left or down in the entries of the temporator matrix. "=" can be treated as "$\le$" or "$\ge$", depending on the shift direction. The result adds to the current symbol in the correspnding fields, i.e. the logical operation is logical "and" or conjunction.

Depropagation means to replace all symbols contained in a temporator by the symbol "$\cong$", without changing the overall information of the temporator. Here, as many symbols "$\cong$" as possible should be introduced.

Depropagation ist a necessary precondition for temporator multiplication which will be explained in the next section. It is the opposite operation to propagation, replacing $\left\langle\begin{array}{c|c} < & \cong \\ \hline < & \cong \end{array}\right\rangle$ by $\left\langle\begin{array}{c|c} \cong & \cong \\ \hline < & \cong \end{array}\right\rangle$ etc. where this replacement does not change the meaning of the expressions.

Propagation is an easy, systematic way to check properties of temporators. Although it looks awkward at the first glance, it helps to reveal many properties of relationships.

In literature the following simple temporators are considered

	strongest	weakest	
meaning	temporator		interval relationship
A before B	$A\left\langle\begin{array}{c\|c} < & < \\ \hline < & < \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} \cong & \cong \\ \hline < & \cong \end{array}\right\rangle B$	|——A——| |——B——|
A after B	$A\left\langle\begin{array}{c\|c} > & > \\ \hline > & > \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} \cong & > \\ \hline \cong & \cong \end{array}\right\rangle B$	|——B——| |——A——|
A equals B	$A\left\langle\begin{array}{c\|c} = & \le \\ \hline \ge & = \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} = & \cong \\ \hline \cong & = \end{array}\right\rangle B$	|——A——| |——B——|
A at B	$A\left\langle\begin{array}{c\|c} \le & \le \\ \hline = & \le \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} \cong & \cong \\ \hline = & \cong \end{array}\right\rangle B$	|——A——|——B——|
B at A	$A\left\langle\begin{array}{c\|c} \ge & = \\ \hline \ge & \ge \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} \cong & = \\ \hline \cong & \cong \end{array}\right\rangle B$	|——B——|——A——|
A overlays B	$A\left\langle\begin{array}{c\|c} < & < \\ \hline > & < \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} < & \cong \\ \hline > & < \end{array}\right\rangle B$	|——A——| |——B——|
B overlays A	$A\left\langle\begin{array}{c\|c} > & < \\ \hline > & > \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} > & < \\ \hline \cong & > \end{array}\right\rangle B$	|——A——| |——B——|
A during B	$A\left\langle\begin{array}{c\|c} > & < \\ \hline > & < \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} > & \cong \\ \hline \cong & < \end{array}\right\rangle B$	|—A—| |——B——|
B during A	$A\left\langle\begin{array}{c\|c} < & < \\ \hline > & > \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} < & \cong \\ \hline \cong & > \end{array}\right\rangle B$	|——A——| |—B—|
A starts B	$A\left\langle\begin{array}{c\|c} = & < \\ \hline \ge & < \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} = & \cong \\ \hline \cong & < \end{array}\right\rangle B$	|—A—| |——B——|
B starts A	$A\left\langle\begin{array}{c\|c} = & \le \\ \hline > & > \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} = & \cong \\ \hline \cong & > \end{array}\right\rangle B$	|——A——| |—B—|
A terminates B	$A\left\langle\begin{array}{c\|c} > & \le \\ \hline > & = \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} > & \cong \\ \hline \cong & = \end{array}\right\rangle B$	|—A—| |——B——|
B terminates A	$A\left\langle\begin{array}{c\|c} < & < \\ \hline \le & = \end{array}\right\rangle B$	$A\left\langle\begin{array}{c\|c} < & \cong \\ \hline \cong & = \end{array}\right\rangle B$	|——A——| |—B—|

From these, other temporators can be constructed easily, for example by substituting "≤" by "<". In the third row we have given the weakest temporator, from which the strongest temporators can be derived. A symbol is called "strong" if it cannot be derived by any other symbol in the considered temporator; otherwise it is called "weak". Thus propagation means to insert weak symbols; de-propagation means to remove weak symbols.

We have estimated the number of different consistent temporators, which are independent and consistent. This has been done by a computer program that propagated all $7^4 = 2401$ different temporators and compared them with all others. This computation resulted in 299 different consistent temporators.

3 Multiplication of temporators

This section is central to the previous discussion of elementary results of temporators as well as to the next section, where quantitative temporators are introduced. In this section we solve the problem to compute for two temporators α and β, for which hold

$A\,\alpha\,B$ and $B\,\beta\,C$,

a temporator, for which holds

$A\,\gamma\,C$.

This means, we have to find γ for which the logical implication

$$A\,\alpha\,B \quad \wedge \quad B\,\beta\,C \quad \Rightarrow \quad A\,\gamma\,C$$

holds, where γ should enclose all information that can be derived from $A\,\alpha\,B$ and $B\,\beta\,C$.

The solution to this problem can be given by explicitly stating all relationships between all points A_i and C_j, deriving the strongest relationship from this and writing this into each field $\left\langle \frac{\circ}{\circ} \middle| \frac{\circ}{\circ} \right\rangle$ of temporator γ. This method is cumbersome and will therefore be substituted by a simple algorithmic solution.

Let us at first look at the explicit solution. This would state the relationships between

$$(A_1,A_2),\ (A_1,B_1),\ (A_1,B_2),\ (A_2,B_1),\ (A_2,B_2),\ (B_1,B_2),$$
$$(B_1,C_1),\ (B_1,C_2),\ (B_2,C_1),\ (B_2,C_2),\ (C_1,C_2).$$

and derive the missing relationships

$$(A_1,C_1),\ (C_1,C_2),\ (A_2,C_1),\ (A_2,C_2)$$

from transitivity conclusions like

if $A_1 < B_1$ and $B_1 < C_1$ then $A_1 < C_1$.

Using B$_2$ instead of B$_1$ there might be another relationship between (A_1,C_1), derivable by

if $A_1 \le B_2$ and $B_2 = C_1$ then $A_1 \le C_1$,

which generally can be weaker or stronger than the first one. Since both relationships must hold, the strongest one should be used as relationship between (A_1,C_1).

"Stronger" means: "less possibilities", i.e. "<" is stronger than "$\le$", or "=" is stronger than "$\ge$" etc. The following table formalises the notion of stronger.

$\oplus$	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$	$\cong$
$<$	$<$	$<$	$\varnothing$	$<$	$\varnothing$	$\varnothing$	$<$
$\leq$	$<$	$\leq$	$=$	$<$	$=$	$\varnothing$	$\leq$
$=$	$\varnothing$	$=$	$=$	$\varnothing$	$=$	$\varnothing$	$=$
$\neq$	$<$	$<$	$\varnothing$	$\neq$	$>$	$>$	$\neq$
$\geq$	$\varnothing$	$=$	$=$	$>$	$\geq$	$>$	$\geq$
$>$	$\varnothing$	$\varnothing$	$\varnothing$	$>$	$>$	$>$	$>$
$\cong$	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$	$\cong$

The operator symbol $\oplus$ is used to estimate from two symbols the stronger one. To formalise transitivity we can use the following table. The corresponding operator is displayed by $\otimes$.

$\otimes$	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$	$\cong$
$<$	$<$	$<$	$<$	$\cong$	$\cong$	$\cong$	$\cong$
$\leq$	$<$	$\leq$	$\leq$	$\cong$	$\cong$	$\cong$	$\cong$
$=$	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$	$\cong$
$\neq$	$\cong$	$\cong$	$\neq$	$\cong$	$\cong$	$\cong$	$\cong$
$\geq$	$\cong$	$\cong$	$\geq$	$\cong$	$\geq$	$>$	$\cong$
$>$	$\cong$	$\cong$	$>$	$\cong$	$>$	$>$	$\cong$
$\cong$	$\cong$	$\cong$	$\cong$	$\cong$	$\cong$	$\cong$	$\cong$

Now the algorithm to compute temporator γ, when α and β are given, should be clear. First compute the "product" of the line of the first temporator with the column of the second temporator according to table "$\otimes$"; then compute the "sum" of these products according to table "$\oplus$". The sequence of operations is exactly the same as the sequence of multiplication of matrices in linear algebra. Thus if

$$A\left\langle \frac{a}{c}\middle|\frac{b}{d}\right\rangle B \text{ and } B\left\langle \frac{e}{g}\middle|\frac{f}{h}\right\rangle C$$

are given, we compute

$$\gamma = \left\langle \frac{(a\otimes e)\oplus(b\otimes g)}{(c\otimes e)\oplus(d\otimes g)}\middle|\frac{(a\otimes f)\oplus(b\otimes h)}{(c\otimes f)\oplus(d\otimes h)}\right\rangle$$

to get γ. This shows that there is a (relatively) simple algorithm to compute new relationships from given ones.

This method can be extended to multiple-intervals as well. Since the number of points of B determines the number of columns in α and the number of rows in β, the corresponding rules for matrix-multiplication hold also. The result is a relationship between intervals A and C, according to the dimensions between these intervals. Thus multiple-intervals can be treated exactly in the same way as 2-intervals.

Examples

Let B start after the end of A, and C start after the end of B. Then we get the temporators $A\left\langle\frac{\cong}{<}\middle|\frac{\cong}{\cong}\right\rangle B$ and $B\left\langle\frac{\cong}{<}\middle|\frac{\cong}{\cong}\right\rangle C$, or after propagation follows

$$A\left\langle\frac{<}{<}\middle|\frac{<}{<}\right\rangle B \text{ and } B\left\langle\frac{<}{<}\middle|\frac{<}{<}\right\rangle C \Rightarrow A\left\langle\frac{<}{<}\middle|\frac{<}{<}\right\rangle C \Leftrightarrow A\left\langle\frac{\cong}{<}\middle|\frac{\cong}{\cong}\right\rangle C\,.$$

Thus C starts after the end of A, which is not very surprising. However, this statement is computed automatically, without any "considerations about the meaning" during computation. Thus this algorithmic method can be executed on computers.

Now let B start in A and B start after C. Then follows

$$A\left\langle\frac{<}{>}\middle|\frac{<}{<}\right\rangle B \wedge B\left\langle\frac{>}{>}\middle|\frac{>}{>}\right\rangle C \Rightarrow A\left\langle\frac{\cong}{>}\middle|\frac{\cong}{>}\right\rangle C.$$

Thus A terminates after C.

The next example uses four intervals, where we have the conditions

- A starts within B,
- B does not start before C,
- B starts after D,
- C and D start at the same time.

We formalise these requirements, which makes the statements more precise, since our verbal definition is not quite unique, and compute

$$A\left\langle\frac{>}{\cong}\middle|\frac{<}{\cong}\right\rangle B \wedge B\left\langle\frac{\geq}{\cong}\middle|\frac{\cong}{\cong}\right\rangle C \wedge B\left\langle\frac{\cong}{\cong}\middle|\frac{>}{\cong}\right\rangle D \wedge D\left\langle\frac{=}{\cong}\middle|\frac{\cong}{\cong}\right\rangle C \Leftrightarrow$$

$$A\left\langle\frac{>}{>}\middle|\frac{<}{\cong}\right\rangle B \wedge B\left\langle\frac{\geq}{\geq}\middle|\frac{\cong}{\cong}\right\rangle C \wedge B\left\langle\frac{>}{>}\middle|\frac{>}{>}\right\rangle D \wedge D\left\langle\frac{=}{\geq}\middle|\frac{\leq}{\cong}\right\rangle C \Leftrightarrow$$

$$A\left\langle\frac{>}{\cong}\middle|\frac{\cong}{\cong}\right\rangle C \wedge A\left\langle\frac{>}{\cong}\middle|\frac{<}{\cong}\right\rangle B \wedge B\left\langle\frac{>}{>}\middle|\frac{\cong}{\cong}\right\rangle C \Rightarrow A\left\langle\frac{>}{\cong}\middle|\frac{\cong}{\cong}\right\rangle C\,.$$

The first equivalence makes the propagation, the second one multiplies the first and the second temporators, as well as another multiplication, which however does not yield additional information. The result is that A starts after C.

The examples from above show, that complex semantics is to be expressed with complex syntax. It should be easy to simplify notation in some senses. We will in the following examples drop the symbol „$\cong$“ in a temporator (and only there), since it is always possible to recognise the correct meaning of a temporator.

Some requirements are given in a negative form, so that negation of a temporator is an important operation. If there is only one symbol within the temporator (besides "$\cong$") then negation means to replace this symbol by its negative meaning, which follows from the next table.

	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$	$\cong$	$\varnothing$
$\neg$	$\geq$	$>$	$\neq$	$=$	$<$	$\leq$	$\cong$	$\varnothing$

The last two symbols follow from the fact that "$\cong$" means no restriction, and "$\varnothing$" means impossible. A temporator with a "$\varnothing$" should not be treated anyway, since it expresses inconsistency, while "$\cong$" means that any relationship is possible, the negation of which is still any relationship.

If there are several symbols, and if it is not possible to depropagate them, simple negation for each field has to be done and then a disjunctive combination of the corresponding fields must be performed. For example: „A does not start B“ means: $\neg\left(A\ \textit{starts}\ B\right)$, or

$$\neg\left(A\left\langle\begin{array}{c|c} = & \\ \hline & < \end{array}\right\rangle B\right) \Leftrightarrow \neg\left(A\left\langle\begin{array}{c|c} = & \\ \hline & \end{array}\right\rangle B \wedge A\left\langle\begin{array}{c|c} & \\ \hline & < \end{array}\right\rangle B\right) \Leftrightarrow A\left\langle\begin{array}{c|c} \neq & \\ \hline & \end{array}\right\rangle B \vee A\left\langle\begin{array}{c|c} & \\ \hline & \geq \end{array}\right\rangle B \Leftrightarrow A\left\langle\begin{array}{c|c} \neq & \\ \hline & \end{array} \vee \begin{array}{c|c} & \\ \hline & \geq \end{array}\right\rangle B$$

This means that either the starting points of A and B are different, or A terminates after or with B. This cannot be expressed by a simple temporator, which shows the limitations of our formalism. Thus we introduce a new notation, using an additional "$\vee$" to express logical "or".

Example

Let us consider two 3-intervals A and B and a 2-interval C. A‘s and B‘s first intervals do not overlay, which is a requirement often found as condition for the initial phase of big electrical machines, since both together would consume too much electrical power or since there are not enough operators available. Also we assume that „A terminate before B“ and „the first phase of B overlay C“. A formal notation looks like this,

$$A\left\langle\begin{array}{c|c|c} & & \\ \hline < & & \\ \hline & & \leq \end{array} \vee \begin{array}{c|c|c} & > & \\ \hline & & \\ \hline & & \leq \end{array}\right\rangle B \text{ and } B\left\langle\begin{array}{c|c} < & \\ \hline & > \\ \hline & \end{array}\right\rangle C$$

Multiplying, we get a relationship between A and C,

$$A\left\langle\begin{array}{c|c|c} & & \\ \hline < & & \\ \hline & & \leq \end{array} \vee \begin{array}{c|c|c} & > & \\ \hline & & \\ \hline & & \leq \end{array}\right\rangle \times \left\langle\begin{array}{c|c} < & \\ \hline & > \\ \hline & \end{array}\right\rangle C \Leftrightarrow A\left\langle\begin{array}{c|c} & \\ \hline < & \\ \hline & \end{array} \vee \begin{array}{c|c} & > \\ \hline & \\ \hline & \end{array}\right\rangle C$$

which states that C is not during A‘s first phase, since we have $A_2 < C_1 \vee A_1 > C_{21}$.

4 Quantitative temporators

Now we introduce quantitative temporators. We start from a quantitative relationship between two points, which is stated as an example,

$$A_1 + 5 < B_2 .$$

This means that B terminates later than 5 units of time after A starts. Using temporators we can write

$$A \overset{+\,5\,<}{\text{—}\!\!\!|\!\!\!\text{—}} B .$$

This means that the „inner“ part of an inequality operation symbol extends to a constant term

$$A_1[+5<]B_2 .$$

Again, this is a generalisation of the classical inequality operation symbol "<" with a constant 0. The sign of the number can also become negative. Of course, for simplicity, the „+“ can be dropped, so we get

$$A \overset{5\,<}{\text{—}\!\!\!|\!\!\!\text{—}} B .$$

We are interested in operations on these temporators. They are again a straightforward extension of the methods of the last section, where now only the constant is to be considered. Thus

$$A_i + x < B_j \wedge B_j + y < C_k \Rightarrow A_i + x < B_j < C_k - y \Rightarrow A_i + x + y < C_k .$$

The conditions are changed by adding the constants; everything else remains the same. For the transitivity relationships we can use the following table.

⊗	+x<	+x≤	+x=	+x≠	+x≥	+x>
+y<	+x+y<	+x+y<	+x+y<			
+y≤	+x+y<	+x+y≤	+x+y≤			
+y=	+x+y<	+x+y≤	+x+y=	+x+y≠	+x+y≥	+x+y>
+y≠			+x+y≠			
+y≥			+x+y≥		+x+y≥	+x+y>
+y>			+x+y>		+x+y>	+x+y>

However, some more changes are to be made by the selection of the "stronger" relationship.

If $\left\{\begin{matrix} a+x \le c \\ a+y \le c \end{matrix}\right\}$ then $a + \max(x,y) \le c$, thus $+x \le \oplus + y \le \Rightarrow + \max(x,y) \le$.

If there is "<" instead of "≤" for the larger value x or y, then the result is $+\max(x,y) <$.

If $\left\{\begin{matrix} a+x \ge c \\ a+y \ge c \end{matrix}\right\}$ then $a + \min(x,y) \ge c$, thus $+x \ge \oplus + y \ge \Rightarrow + \min(x,y) \ge$.

If there is ">" instead of "≥" for the smaller value x or y, then the result is $+\min(x,y) >$

If $\left\{\begin{matrix} a+x=c \\ a+y=c \end{matrix}\right\} \wedge x=y$ then $a+x=c$, thus $+x=\oplus+y=\Rightarrow \left\{\begin{matrix} +x= & if\ x=y \\ \varnothing & else \end{matrix}\right.$.

If $\left\{\begin{matrix} a+x=c \\ a+y\neq c \end{matrix}\right\} \wedge x\neq y$ then $a+x=c$, thus $+x=\oplus+y\neq\Rightarrow \left\{\begin{matrix} +x= & if\ x\neq y \\ \varnothing & else \end{matrix}\right.$.

If $\left\{\begin{matrix} a+x\neq c \\ a+y\neq c \end{matrix}\right\} \wedge x=y$ then $a+x\neq c$, thus $+x\neq\oplus+y\neq\Rightarrow \left\{\begin{matrix} +x\neq & if\ x=y \\ +x\neq\wedge+y\neq & else \end{matrix}\right.$.

If $x\neq y$, then the two conditions cannot be combined, so that both conditions continue to hold. This is also true for the last case as well.

If $\left\{\begin{matrix} a+x\leq c \\ a+y\geq c \end{matrix}\right\} \wedge x=y$ then $a+x=c$, thus $+x\leq\oplus+y\geq\Rightarrow \left\{\begin{matrix} +x\leq\wedge+y\geq & if\ x<y \\ +x= & if\ x=y \\ \varnothing & if\ x>y \end{matrix}\right.$.

This shows that the algorithm is more complex. In spite of this, quantitative temporators are useful for many reasons.

Quantitative temporators can be propagated in a similar way as qualitative temporators. However, now the relationships can be much more differentiated, since the length of an interval A can be described in several ways, like $A_1+4<A_2$, which means that the length of the interval A is more than 4, while $A_1+5\geq A_2$ means that A's length is bounded by 5, since $5\geq A_2-A_1$. In any case, $A_1\leq A_2$ still holds. This can be expressed by an A-A-temporator, like

$$A\left\langle \frac{=}{-4>} \middle| \frac{+4<}{=} \right\rangle A$$

for the first and

$$A\left\langle \frac{=}{-5\leq} \middle| \frac{+5\geq}{=} \right\rangle A$$

for the second example; the symbol "=" holds because $A_1=A_1$ and $A_2=A_2$. However, the second statement ($-5\leq$ in the last example) is completely redundant. Since in any case holds $A_2\geq A_1$, it is more useful to place this statement into the field, which yields

$$A\left\langle \frac{=}{\geq} \middle| \frac{+5\geq}{=} \right\rangle A.$$

Instead of this, one might use the second statement to express more complex information about an interval. It is now possible to combine both expressions in one temporator. If both hold, $A_1+4<A_2$ and $A_1+5\geq A_2$ then $A_1+4<A_2\leq A_1+5$; this confines A's length to the range 4 until 5. This can be written as

$$A\left\langle \frac{=}{-5\le} \middle| \frac{+4<}{=} \right\rangle A.$$

If such temporators are given for some intervals, then propagation takes place by multiplication, since from $A\alpha A$ and $A\beta B$ follows $A\alpha\times\beta B$. We find for our example, assuming that B starts more than one unit of time after A starts,

$$A\left\langle \frac{=}{-5\le} \middle| \frac{+4<}{=} \right\rangle A \text{ and } A\left\langle \frac{+1<}{} \middle| - \right\rangle B \Rightarrow A\left\langle \frac{+1<}{-4<} \middle| - \right\rangle B.$$

Since always holds $B_1 \le B_2$, follows from $A_1+1<B_1\le B_2$ that $A_1+1<B_2$. Thus we get

$$A\left\langle \frac{+1<}{-4<} \middle| \frac{+1<}{-4<} \right\rangle B.$$

Thus also propagation becomes more complex for quantitative temporators than for qualitative ones. If a propagated symbol combines with a resident one, of course the stronger condition is to be taken. In the last example we add the condition that B ends at least 3 units of time after A ends; then follows

$$A\left\langle \frac{=}{-5\le} \middle| \frac{+4<}{=} \right\rangle A \text{ and } A\left\langle \frac{+1<}{} \middle| \frac{}{+3\le} \right\rangle B \Rightarrow A\left\langle \frac{+1<}{-4<} \middle| \frac{+7<}{+3\le} \right\rangle B.$$

This implies after propagation

$$A\left\langle \frac{+1<}{-4<} \middle| \frac{+7<}{+3\le} \right\rangle B \Rightarrow A\left\langle \frac{+1<}{-4<} \middle| \frac{+1<\wedge+7<}{-4<\wedge+3\le} \right\rangle B \Rightarrow A\left\langle \frac{+1<}{-4<} \middle| \frac{+7<}{+3\le} \right\rangle B.$$

So propagation does not change this temporator. However, propagation can become much more complicated. For example we have

$$A\left\langle \frac{a\le}{-c\ge} \middle| \frac{}{b\le} \right\rangle B \Rightarrow A\left\langle \frac{a\le}{-c\ge} \middle| \frac{a+b+c\le}{+b\le} \right\rangle B$$

This follows from $A_1+a\le B_1$, $A_2-c\ge B_1$, and $A_2+b\le B_2$, since now we can conclude

$$A_1+a+b+c\le B_1+b+c\le A_2+b\le B_2.$$

Here all conditions in the assumption are required, particularly $A_2-c\ge B_1$. This shows that propagation is quite more complicated than expected.

Let us now consider an example to apply quantitative temporators.

Example

This example solves a classical scheduling problem. Let us assume there are four jobs given, the relationship of which will be expressed by temporators. Having done this, we will compute further relationships of jobs, which were not stated explicitly. These jobs are called A, B, C and D, respectively, with self-related properties

$$A\left\langle\begin{array}{c|c} = & 5= \\ \hline \geq & = \end{array}\right\rangle A,\ B\left\langle\begin{array}{c|c} = & 6= \\ \hline \geq & = \end{array}\right\rangle B,\ C\left\langle\begin{array}{c|c} = & 4= \\ \hline \geq & = \end{array}\right\rangle C, \text{ and } D\left\langle\begin{array}{c|c} = & 4= \\ \hline \geq & = \end{array}\right\rangle D.$$

Obviously, the length of the four jobs is exactly 5, 6, 4, and 4, respectively. Now we state relationships between jobs. They are

$$A\left\langle\begin{array}{c|c} & \\ \hline \leq & \end{array}\right\rangle B,\ A\left\langle\begin{array}{c|c} & \\ \hline \leq & \end{array}\right\rangle C,\ B\left\langle\begin{array}{c|c} & \\ \hline \leq & \end{array}\right\rangle D, \text{ and } C\left\langle\begin{array}{c|c} & \\ \hline \leq & \end{array}\right\rangle D.$$

Again, these statement are easily to be interpreted. They state that B or C do not start before A ends, and that D does not start before B and C have ended. If we compute the products, we find after propagation

$$A\left\langle\begin{array}{c|c} = & 5= \\ \hline \geq & = \end{array}\right\rangle A,\ A\left\langle\begin{array}{c|c} \leq & \leq \\ \hline \leq & \leq \end{array}\right\rangle B, B\left\langle\begin{array}{c|c} = & 6= \\ \hline \geq & = \end{array}\right\rangle B \Rightarrow A\left\langle\begin{array}{c|c} 5\leq & 5\leq \\ \hline \leq & \leq \end{array}\right\rangle B, B\left\langle\begin{array}{c|c} = & 6= \\ \hline \geq & = \end{array}\right\rangle B \Rightarrow A\left\langle\begin{array}{c|c} 5\leq & 11\leq \\ \hline \leq & 6\leq \end{array}\right\rangle B,$$

$$A\left\langle\begin{array}{c|c} = & 5= \\ \hline \geq & = \end{array}\right\rangle A,\ A\left\langle\begin{array}{c|c} \leq & \leq \\ \hline \leq & \leq \end{array}\right\rangle C, C\left\langle\begin{array}{c|c} = & 4= \\ \hline \geq & = \end{array}\right\rangle C \Rightarrow A\left\langle\begin{array}{c|c} 5\leq & 5\leq \\ \hline \leq & \leq \end{array}\right\rangle C, C\left\langle\begin{array}{c|c} = & 4= \\ \hline \geq & = \end{array}\right\rangle C \Rightarrow A\left\langle\begin{array}{c|c} 5\leq & 9\leq \\ \hline \leq & 4\leq \end{array}\right\rangle C.$$

We see from this example, that information about the system can easily be introduced by „simple“ temporators, where more complex information can be evaluated by an algorithmic method, and so by computers. The next step is now to use information about job D.

$$A\left\langle\begin{array}{c|c} 5\leq & 11\leq \\ \hline \leq & 6\leq \end{array}\right\rangle B, B\left\langle\begin{array}{c|c} \leq & \leq \\ \hline \leq & \leq \end{array}\right\rangle D, D\left\langle\begin{array}{c|c} = & 4= \\ \hline \geq & = \end{array}\right\rangle D \Rightarrow A\left\langle\begin{array}{c|c} 11\leq & 11\leq \\ \hline 6\leq & 6\leq \end{array}\right\rangle D, D\left\langle\begin{array}{c|c} = & 4= \\ \hline \geq & = \end{array}\right\rangle D \Rightarrow A\left\langle\begin{array}{c|c} 11\leq & 15\leq \\ \hline 6\leq & 10\leq \end{array}\right\rangle D$$

The reader is asked to compute the corresponding result with C instead of B and show that he gets $A\left\langle\begin{array}{c|c} 9\leq & 13\leq \\ \hline 4\leq & 8\leq \end{array}\right\rangle D$.

Our result states besides others that $A_1 + 15 \leq D_2$, which means of course that D wont be finished earlier than 15 units of time after A starts.

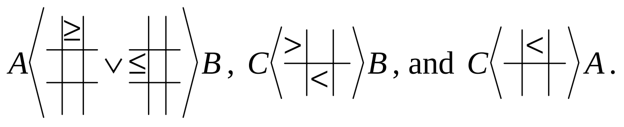
This example shows that quantitative reasoning about intervals can be performed algorithmically, since the notation evaluated in this paper helps to manage the complex information, which is necessary to be handled. However, our method allows to model quite more complex relationships. The next example gives an impression of this.

Example

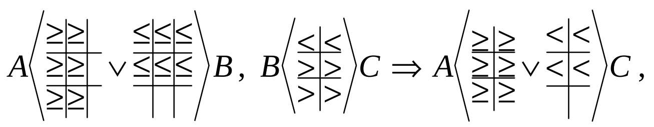
This example continues our former example about multiple-intervals. We assume that there are three intervals, A, B, C, with the following self-relating properties.

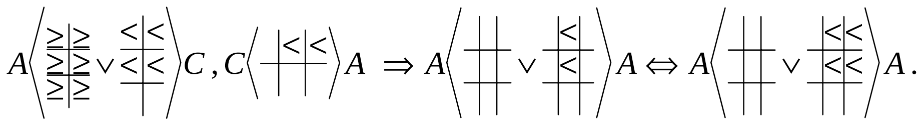
$$A\left\langle\begin{array}{c|c|c} = & 1\leq & \\ \hline & = & 5\leq \\ \hline & & = \end{array}\right\rangle A,\ B\left\langle\begin{array}{c|c|c} = & 2\leq & \\ \hline & = & 6\leq \\ \hline & & = \end{array}\right\rangle B, \text{ and } C\left\langle\begin{array}{c|c} = & 3\leq \\ \hline & = \end{array}\right\rangle C.$$

Again we have to avoid that A's and B's first intervals interfere, and again we demand that C lies completely in the first interval of B. Also, C must start before A's second phase begins.

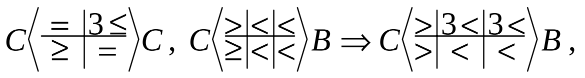


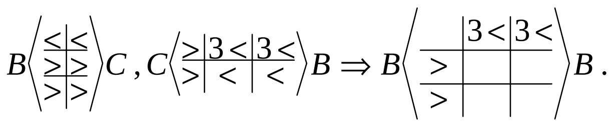
Our first objective is to find another self-relationship for A. Propagating and multiplying yields





The second term of the disjunction contradicts the requirement that $A_2 = A_2$, so that this case can never become true. So we have formally concluded that A must be started after B. Now let us apply a quantitative argumentation.





Comparing this with the original self-relation of B shows that its first phase necessarily is bigger than 3, not 2 as stated above. Again we have derived formally a result that follows from the assumption.

5 Conclusion

The method presented here was developed to state relationships between intervals, which can be applied to many areas of specification techniques. Our approach extends Allen's method to compute new relationships by given ones by a simple "multiplication rule" that can be applied to very general types of intervals, like multi-intervals or quantitative statements about intervals. The main advantage of our method seems to be the possibility to express very complex relationships between intervals in a fairly simply way, which easily can be interpreted. Since the derived symbols express complex meaning they are rather complex.

Future research includes

- development of methods to overcome some limits, we have mentioned,
- implementation of the calculations in a computer program that can
  - handle arbitrary numbers of relationships between intervals,

- evaluate relationships between intervals,
- perform consistency checks,
- perform optimisation of relationships.

It should be mentioned here that our method can also be applied to other models, including Boolean or Aristotelian logic and set theory [Kowalk92].

6 Literature

Allen84	J.F. Allen: "Towards a General Theory of Action and Time" Artificial Intel-legnce 23 (1984), S. 123-154
Kowalk92	W. P. Kowalk: Konstruktorentechnik. Bericht aus dem FB Informatik der Universität Oldenburg. (Okt. 1992), Nr. 5/92.

7 Methodological Considerations

Now, we give a short look back to the method we have introduced. At first, we mention that similar systems can be applied to other objects as well, such as set theory, formal logic (Boolean or "Aristotelian"), or can be used as operators resulting in another object, like set operators. All these systems need to describe relationships between objects, which are here intervals, or which can be sets (or subsets), or statements on logical assertions etc. In all cases, the statements were constructed by use of matrices, describing the corresponding relationships by some more basic relationships. Because of the methodological approach, we call this "construction techniques", the symbols "constructed operators" or "constructors".

This method is very common in the most technical sciences. Instead of using some basic statements, called axioms, and derive some properties from them, we use a very big set of different symbols to describe the relationships or operations. In all cases it is possible to construct new true statements by some multiplying operations on these symbols, which where all very similar to matrix multiplication as we have shown here.

The gain from this method is obvious. The task is no longer to prove some properties or relationships by looking for a derivation from a system of axioms, we can now "compute" such relationships, with little effort for man or computer. The results – since derived by an algorithm – are usually correct. Also, this method is much more flexible than many other methods. Introduction of additional features like multiple-intervals or quantitative relationships can be added without big problems, and the argumentation is very similar to the simpler system from which we have derived our method, which is here the simple theory of inequality.

Computer scientists should therefore be careful to adopt too careless methods of other disciplines; particular mathematical methods and techniques are usual too "over formalised" and cumbersome, which is not really required in a practical, application oriented science like computer science.

8 Consistency Check

A severe problem for time logic is consistency. Given a set of intervals $\{A, B, \ldots\}$ and some temporators stating relationships between them, one might ask whether the relationships are consistent, i.e. there is at least one concrete set of interval borders, so that all relationships between all intervals can be fulfilled at the same time. This will be called consistency requirement. It is for example a typical task in schedule problems, where the existence of a solution is asked for. We consider in this section how this problem might be solved.

If a set of time periods $\{A, B, \ldots\}$ is given, together with some temporators stating relationships between them, we can represent them as a set of relationships between the endpoints of the intervals. This is always possible, even if some of them are multiple-intervals. For each 2-2-temporator we find at most six such relationships, including those two stating evident relationships between the left and right points of those intervals. Thus the number of such relationships is limited to the number of temporators times a constant factor. It should be obvious that all information that can be stated about such intervals is given in the limited number of temporators. Thus no derivation of any further relationships between the intervals by our methods is required; one should us the „simplest“ ones, i.e. apply depropagation, wherever possible. Also, if quantitative temporators are used, the relationship is simply extend by a constant sum.

We identify points that must be equal "=", ignore relationships that state inequality "$\neq$", replace "<" by "$\leq$" and ">" by "$\geq$", and sort the residual set of relationships according to a topological sorting algorithm.

Such a topological algorithm counts for each point X the number of relationships $X + a \leq Y$ which we call N_X. For all X, for which $N_X = 0$, X is "smallest" point and can be numbered 1, 2, … For all such smallest points X we decrement N_Z for all relationships $X \leq Z$. If $N_Z - 1$ becomes 0, we perform in the next cycle the same algorithm for this Z, determining further points W, where $N_W - 1$ becomes 0 etc. The result is a topological sorting of the points, where the conditions $Y \leq X$ hold, provided they are required. If however some points W are left, where N_W can’t be made to zero by this algorithm, the set of temporators is inconsistent.

This is an efficient algorithm that can be used to solve the problem to check consistency of a set of qualitative and quantitative temporal relationships. Although algorithms to test this are still under development, this might help so solve problems in areas as production planning systems or task scheduling.