Temporal representation and relations in OpenCog

Dario Garcia-Gasulla

June 26, 2012

1 Basic representation

We intend to build a fuzzy interval representation of time and use Allen's temporal algebra to reason with it. Considering the current representations of time in OpenCog, which are implemented partially with timestamps, to define a period of time we will use a list of timestamps each of which has an associated certainty that defines the degree at which the period takes place at that timestamp. That list of timestamps will define a function over time computing the certainty of the interval.

Allen's temporal relations between two elements, A and B, can be computed by making comparisons between *"the beginning or end of A"* against *"the beginning or end of B"* (from now on, a^- represents the beginning of A, a^+ represents the ending of A). For example, A happens before B if and only if a^+ happens before b^-. See Figure 1 for details on the rest of relations.

Accordingly, we can implement Allen's temporal relations by first implementing the comparison of its sub-intervals. Once having that, the formulas calculating the fuzzy temporal relations are easy to implement with simple fuzzy logic operators.

A first step is therefore to define what is a^+ and a^- w.r.t. our representation of a fuzzy temporal interval. We have decided to define the sub-intervals of timestamps as follows:

Definition 1. *Given an ordered list of pairs* $< timestamp, certainty >$ *T, we define:*

- *The beginning of T (t^-) as the sub-intervals containing all the timestamps between the first element in which the certainty is not 0 and the first element in which the certainty is 1, both included.*
- *The end of T (t^+) as the sub-intervals containing all the timestamps between the first element (starting from the latest) in which the certainty is not 0 and the first element (starting from the latest) in which the certainty is 1, both included.*

Following this definition, lets consider an example. If the distribution of A was defined by the following timestamps:

- timestamp1, certainty 0

Allen's Temporal Interval Relations Between Intervals
$A = [a^-, a^+]$ and $B = [b^-, b^+]$

Name	Definition		
before	$b(A,B)$	$\equiv$	$a^+ < b^-$
overlaps	$o(A,B)$	$\equiv$	$a^- < b^-$ and $b^- < a^+$ and $a^+ < b^+$
during	$d(A,B)$	$\equiv$	$b^- < a^-$ and $a^+ < b^+$
meets	$m(A,B)$	$\equiv$	$a^+ = b^-$
starts	$s(A,B)$	$\equiv$	$a^- = b^-$ and $a^+ < b^+$
finishes	$f(A,B)$	$\equiv$	$a^+ = b^+$ and $b^- < a^-$
equals	$e(A,B)$	$\equiv$	$a^- = b^-$ and $a^+ = b^+$
after	$bi(A,B)$	$\equiv$	$b(B,A)$
overlapped-by	$oi(A,B)$	$\equiv$	$o(B,A)$
contains	$di(A,B)$	$\equiv$	$d(B,A)$
met–by	$mi(A,B)$	$\equiv$	$m(B,A)$
started–by	$si(A,B)$	$\equiv$	$s(B,A)$
finished–by	$fi(A,B)$	$\equiv$	$f(B,A)$

Figure 1: Allen's temporal relations reduced to comparisons of sub-intervals. *Source: "Fuzzifying Allen's Temporal Interval Relations", Schockaert et. al. 2008*

- timestamp2, certainty 0.2
- timestamp3, certainty 0.5
- timestamp4, certainty 1
- timestamp5, certainty 1
- timestamp6, certainty 1
- timestamp7, certainty 0

a^- would be the list $< timestamp2, timestamp3, timestamp4 >$, and a^+ would be the list $< timestamps6 >$.

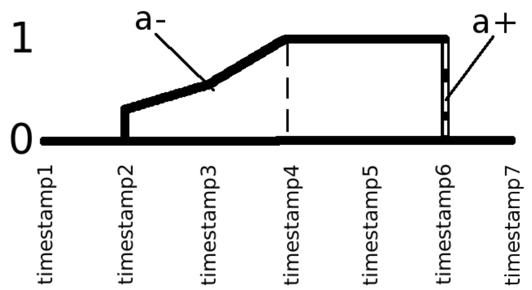

Figure 2: Fuzzy temporal interval A, and its sub-intervals

In code, to represent the complete fuzzy temporal interval one could use a list, a set or a dictionary of pairs $< Timestamp, TruthValues >$.

However, if the data structured is not ordered, it will have to be ordered when calculating its relations.
Even though the sub-intervals representing a^- and a^+ could have many elements (depending on the temporal distribution), we have decided to reduce that set to its first and last elements (*i.e.*, both limits of the sub-interval). By doing so we can obtain a linear function which roughly describes a^- and a^+ by approximation. For example, in the interval of Figure 2 one would consider only timestamp1 and timestamp3 to define the beginning of the interval. An example of the resulting functions can be seen in Figure 3 [1]. Algorithmically speaking, regardless of the containing data structured, with a single iteration through all the pairs $< Timestamp, TruthValues >$ one can guarantee to obtain the required four elements which define the two linear functions.

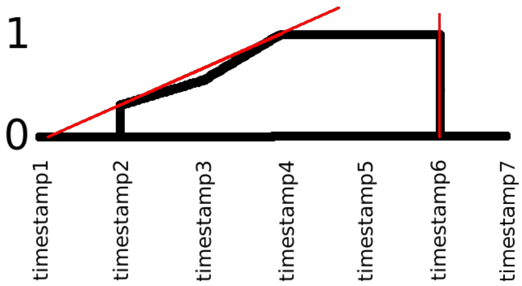

Figure 3: Fuzzy temporal interval A, in red, linear function defining its sub-intervals

2 Simple cases

At this point, assuming that we have the four pairs $< Timestamp, TruthValues >$ defining the beginning and end of the two fuzzy time interval which we want to compare, we can identify some cases in which we can easily know the result of the comparison function:

- If the first of a^+ (*e.g.*, timestamp6) is bigger than the last of b^-, then we can say that $a^+ > b^-$.
  In this case there is a complete overlap, *i.e.*, there's a timestamp which belongs to A with certainty 1 which also belongs to B with certainty 1. Therefore we can be sure that a^+ is bigger than b^-.

- If the first of a^+ and the last b^- are equal and a^+ or b^- is composed by more than one timestamp, then we can say that $a^+ > b^-$.
  In this case there is a complete overlap, *i.e.*, there's a timestamp which belongs to A with certainty 1 which also belongs to B with certainty 1. Therefore we can be sure that a^+ is bigger than b^-.

[1] By reducing the sub-intervals to two points, precision will be lost. However, for the methodology to calculate the comparison between sub-intervals (which we will see in §3), efficiency will be greatly increased since we will be able to work with linear functions.

Figure 4: Complete overlap. $a^+ > b^-$

- If both a^+ and b^- are composed by a single timestamp and it is the same, then we can say that $a^+ = b^-$.
  In this case both sub-intervals meet and are equal. Therefore we can be sure that a^+ is equal to b^-.[2]

`In code, calculating this cases is quite straight forward. Once we have the four pairs` $<Timestamp, TruthValues>$ `stored, we can know whether we are within one of these cases by making simple comparisons of the associated timestamps.`

[3]

3 Complex cases

Beyond those simple cases, it gets more complicated since there is a partial overlap between both sub-intervals. And the properties of that overlap (*e.g.*, how big is it, and where it is located w.r.t. both sub-intervals) will tell us to which degree a^+ is smaller or bigger than b^-.

In our case, we simplified a^+ and b^- to linear functions, so that this overlap is easier to compute. Accordingly we must now compute the degree of overlap from the particularities of both functions. The approximation we have chosen is to take into account only the cross-point of those functions, see Figure 5 for details. [4]

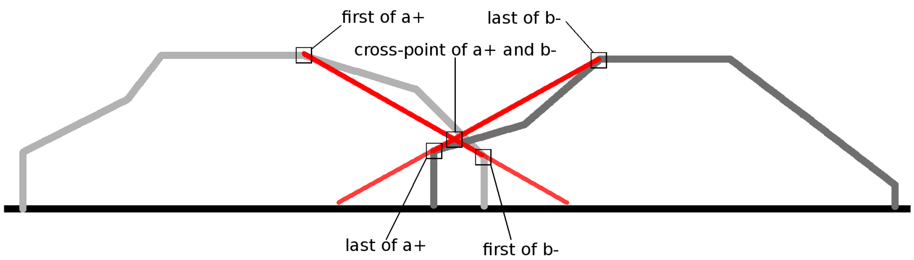

Figure 5: In red: Linear functions representing a+ and b-

[2] I'm not sure about this one, but it is the simpler way of representing $a^+ = b^-$. The other method for representing $a^+ = b^-$, which will be seen in §3, is more analytical and harder for humans to code on purpose.

[3] An alternative to everything that follows is to consider that whenever there is a partial overlap, a^+ is smaller and bigger than b^- with certainty 0.5. It may not be worth, since its quite simple to calculate it with the method proposed in this Section.

[4] Probably a better approximation would be to calculate the center of mass of each sub-interval, and compare their timestamp location. However I cannot think of any cheap way to compute that for the case of the linear functions, even less for the non-linear representation of the sub-intervals.

In what follows, we assume the following definition:

Definition 2. *Given $a^+ < b^-$ with an associated certainty of X, then the certainty of $a^+ > b^-$ is 1-X.*

A good approximation for knowing the relevance of the overlap is to consider the height of the cross point (see Figure5). The higher the cross-point between both sub-intervals is, the highest certainty a shared timestamp has, and the closer the body of both intervals are. Considering that, we define the remain comparison cases as follows:

- If the height of the cross-point is negative or zero, $a^+ < b^-$ with certainty 1. In this case, there is no overlap at all since there is no timestamp with certainty bigger than zero which belongs to both A and B.
- If the height of the cross-point is X, $0 < X < 0.5$, $a^+ < b^-$ with certainty 1-X.
- If the height of the cross-point is X, $X > 0.5$, $a^+ > b^-$ with certainty X.[5]

As an example, consider Figure 5, in which the cross-point is below 0.5 and as a result $a^+ < b^-$ with certainty bigger than 0.5.

`In code, implementing that comparison is relatively easy since we have two linear functions (each one defined by a pair of` $<Timestamp, TruthValues>$`) and we just have to find one coordinate of the cross-point (`*i.e.*`, the certainty of the crossing point). All we have to do is to apply the appropriate formula, which is:`

$$height = -\frac{first\_of\_b^- - last\_of\_a^+}{last\_of\_b^- - first\_of\_b^- + last\_of\_a^+ - first\_of\_a^+}$$

`And we obtain the height of the cross-point. This mathematical operation is quite simple, and after it with a couple of comparisons we can then obtain the degree at which` a^+ `is smaller or bigger than` b^-

Once having that comparison function, all left to do is to implement the AND's operators with fuzzy logic, and we will have a fuzzy representation and a working fuzzy temporal relationship.

Finally, let's consider a special comparison case which is required for some temporal relations (*i.e.*, meets, starts, finishes and equal), and that is the equality between a^+ and b^-. The approach chosen here is to consider that, if $a^+ < b^-$ with certainty 0.5 (and consequently $a^+ > b^-$ with certainty 0.5) then $a^+ = b^-$ with certainty 1. Furthermore, if we assume that if $a^+ < b^-$ with certainty 1 then $a^+ = b^-$ with certainty 0, we could produce the following rule:

Definition 3. *Given $a^+ < b^-$ with certainty X, $a^+ = b^-$ happens with certainty* $1 - |X - 0.5| * 2$

In this case, the certainty of $a^+ < b^-$ and the certainty $a^+ = b^-$ are inversely proportional.

[5]It may be necessary to calculate either X or 1-X depending on the temporal relationship being implemented

4 Discussion

Several points remain to be validated:

- The proposed representation of sub-intervals requires that all fuzzy temporal intervals include a timestamp with certainty = 1. Even though that's relatively easy to guarantee in code (increase the certainty of the timestamp with a highest certainty to 1), does it make sense to do so? Should we be able to deal with intervals which do not satisfy that restriction? An interval which never reaches full certainty is easy to imagine.

- For simplicity reasons, we reduce a^+ and a^- to the first and last timestamp defining them. That allows us to easily calculate the crossing point afterward, and its relevance. Is that reduction worth it? How much precision is lost? If the shape of a^+ and a^- changes a lot (an interval which, after getting a timestamp with certainty 1, drops to 0 several times never reaching 1 again) we may loss relevant data.

- We consider the height of the crossing point as the unique value to calculate the certainty of the comparison. Could we prove the validity of such assumption? In all of the cases I've tested it seems appropriate, but that's a rather weak argument.

- The special case in which a^+ and b^- both are composed of a single timestamp which is the same, is tricky. If someone intends to define two intervals such that $a^+ = b^-$, most likely their first intention will be to codify them as this special case. Although the analytical method explained in §3 makes more sense, it is harder to manually codify an equality according to it. [6] A solution would be to consider this special case as $a^+ = b^-$, as proposed here. Also, if we do not do so, I'm not really sure what to do of this special case.

- I still have to decide which AND fuzzy operator will I use.

[6] Not that hard really, its enough to do $first\_of\_b^- = first\_of\_a^+$ and $last\_of\_a^+ = last\_of\_b^-$