Representation of Continuous Change with Discrete Time

Fernando Barber, Salvador Moreno
Dpto. Electrónica e Informática
University of Valencia
Spain

E-mail: Fernando.Barber@uv.es

Abstract

In this paper we study one of the drawbacks that have been considered against the use of discrete time in Temporal Reasoning Systems, which is the difficulty to represent continuous change. We will study the representation of continuous change for the particular case of a constraint solver over integers. We have used two different approaches. The first approach using the concept of error and the second one using the concept of granularity.

1.-Introduction

In order to represent time in Temporal Reasoning Systems, it is necessary to choose among several possibilities (time points or time intervals, linear time or branching time, discrete time or continuous time, ...). We will consider the choice of discrete time versus continuous time.

Initial formalisms of temporal reasoning adopted discrete time because of its easier use, like the Situation Calculus [8]. Although this system has been very influential on later formalisms, it has many limitations, most of them are due to the use of a discrete representation of time. One of these limitations is the impossibility to represent continuous change.

Hayes [4] introduced the notion of histories, taken from the field of Naive Physics, to overcome these limitations. Also McDermott [9] proposed a temporal logic that solves some of the problems. Both formalisms assume that time is continuous, allowing therefore the representation of continuous change. McDermott, in fact, included in his formalism the representation of continuous change.

Many other formalisms of temporal reasoning have also been augmented to include continuous change as for example the Event Calculus [13], or indeed a new formalization of the Situation Calculus [12] that allows the use of real numbers.

In this paper we study the representation of continuous change using a discrete representation of time. The aim is not to find a better representation or a more efficient representation than using continuous time, but just to show that it is possible to represent continuous change with discrete time and also that it may be efficient. So the decision between the use of discrete or continuous time is not forced by the presence of continuous change in the problem.

The final motivation we have for the use of discrete time is the integration of continuous change inside a constraint solver over finite domains [5] to obtain an efficient management of the continuous change.

We will present two different approaches to the representation of continuous change, the first approach using the concept of error and the second one using time granularity.

2.-Continuous Change

Continuous change [9] is an important issue in temporal reasoning, although many temporal reasoning systems have not considered it. It is fundamental for a realistic representation of time, as can be seen in problems like filling a tank of water or representing the motion of a billiard ball.

We understand as continuous change the continuous variation of a quantity with respect to time. It can be represented by a continuous function of time. We use the term continuous with its mathematical sense, i.e. the variation in the function is not sudden. In this paper we will only consider linear functions of time.

A problem which can be represented with continuous change is the filling of a tank with water. For example, let's suppose we have a tank with a tap over it. The tank has a capacity of 23 litres and the tap has a flow of 4 litres per second. We represent the change in the volume

0-8186-7937-9/97 $10.00

by means of a *trajectory* [13] a function dependent of time that represents the change: $V(t) = 4 \cdot t + Vo$. With the trajectory it is possible to know the volume of the tank at any time, and also to predict at what time will the water reach a given volume.

Continuous change has also been modelled qualitatively [13] [1] and has been also generalised to differential equations [10].

3.-Discrete Time versus Continuous Time

Some formalisms [7][14] avoid the choice between discrete time and continuous time; they use the concept of symbolic times. This is possible as these formalisms do not assume any metric on time, it is only necessary to suppose an order relation between different times. It has the disadvantage that it is impossible to use distances between different time points.

It is possible to go further using the metric of the integer or real numbers, as they are very similar, so we can talk about distances before distinguishing between integer or real numbers. But this distinction is necessary when we state the characteristics of open or closed intervals, and also if we want to represent continuous change. We will concentrate in the choice of discrete versus continuous time for the representation of continuous change.

- **Continuous time** is generally represented using real numbers. Real numbers provide all the possibilities of the real mathematics, the one we use normally in all our problems. The real numbers have two main properties that differentiate them from the integer or rational numbers: density and completeness. The property of density means that between any two points, we can find always another; the property of completeness means that if a series of points is bounded from above, there exists a point that is the least upper bound of that series. Density distinguishes both reals and rationals from the integers, completeness distinguishes the reals from the others.

The main advantage of this representation is that the reals are the numbers used in physics to represent time, so it is easier to base our theories over physical laws, and also the real numbers are a superset of the integer numbers, so they are the most general and appropriate to represent time, but they have also some disadvantages. One disadvantage is the difficulty to make theories with them, due in part to the property of density, which introduces the concept of infinity also in a bound interval. Another disadvantage is that we are not able to represent reals, specially in computers; the usual way to represent them is by means of floating point numbers that are an approximation to reals. This brings more problems like whether two numbers are equal or not (they may be different approximations of the same number), etc...

- **Discrete time** is generally represented using integer numbers. The integer numbers are not commonly used in physics, but they are more intuitive than real numbers, we generally think using integer numbers, so they are more appropriate for a common-sense reasoning. The integer numbers do not have the property of density, this means that between two points there may be no more points.

The main advantage of this representation is that a theory based on integers is easier to make, it use to be more simple. Also the integer numbers have a direct and exact representation over computers, we do not have to work with approximations.

The main disadvantage is that they have some lacks in the representation, specially when we want to represent continuous change as integers cannot represent the solution of an equation system if this solution is fractional. For example, let's take the previous example of the tank. If the tank is empty and we open the tap, when will the water fill the tank?.

The function that represents the change in the volume is $V(t) = 4 \cdot t + Vo$ and the capacity of the tank is 23 litres so we obtain the equation: $4 \cdot t = 23$. The solution of this equation using reals is very clear, $t = 5.75$ s, but if we try to use integers we see that there is no solution, from where it can be deduced that it will be never filled, which is obviously wrong.

We will see in the following sections two different approaches to allow the representation of continuous change (i.e. continuous functions) using discrete time.

4.-An Approach based on Errors

The first way to solve the continuous representation problem using a discrete time representation is to allow an error in the equality of the equations. Let's illustrate it by taking the above example, we have the equation:

$4 \cdot t \cong 23$ so we obtain $t \cong 5$ and $t \cong 6$

The error can be represented inside the equation with a new variable (Err) that in general will follow the inequation:

$|Err| <= MaxError$

Therefore, the equation can be rewritten as:

$4 \cdot t = 23 + Err$

A problem which arises here is the criteria for the value of MaxError. If we choose a little error, it is possible to obtain no solutions; but if we choose a big

error, it can allow a lot of solutions, which has no sense as the solution is unique.

A good criterion is to obtain only one solution, the one nearest to the real solution. But, in general, it is not possible to set the value of MaxError so we obtain only one solution. The conditions to obtain a unique solution are that the errors in each variable must be less than one half and that for a given variable, its error must be the same in any appearance of the variable. These conditions do not let us to represent the criterion in just one equation.

Let's represent a general linear equation as:

$C_1X_1 + \cdots + C_nX_n = C$

The correct way to obtain always one unique solution is to create one error variable for each independent variable and to set the following equations:

$C_1(X_1 + E_1) + \cdots + C_n(X_n + E_n) = C$

$$-\frac{1}{2} \le E_i < +\frac{1}{2} \qquad i = 1..n$$

But we are duplicating the number of variables and setting much more equations. Also, we are using non-integer variables for the errors, which make difficult the uses of these equations with finite domains.

We find in the area of Integer Programming many methods to solve these kind of systems of equations [11], but they are not practical for an incremental set of equations.

It is possible to obtain a more efficient version relaxing the second condition. In this case we can represent the errors in just one equation:

$C_1(X_1 \pm 1/2) + \cdots + C_n(X_n \pm 1/2) = C$

$$C_1X_1 + \cdots + C_nX_n + \frac{\pm C_1 \pm \cdots \pm C_n}{2} = C \qquad (1)$$

$$MaxError = \frac{1}{2}\sum_i |C_i|$$

This value for MaxError guarantees that the equation will always give one solution, if it exists, but in general it will give more than one solution.

In any case, we can use this value of error for practical purposes, as for example a partial inference rule for a constraint solver, with the knowledge of its limitations.

The equation can be introduced in a constraint solver over finite domains as the constraints:

$$C_1X_1 + \cdots + C_nX_n < C + \frac{C_1 + \cdots + C_n}{2}$$
$$C_1X_1 + \cdots + C_nX_n \ge C - \frac{C_1 + \cdots + C_n}{2} \qquad (2)$$

The approach we have presented resolves the problem of representing continuous change, but it has a disadvantage because we still have the theoretical basis of the reals behind this approach. We talk about errors from the real result.

We will see it now from a different point of view, which is granularity.

5.-An Approach based on Granularity

In this approach we will consider integers as an abstraction from the domain of time. This kind of abstraction is known commonly as granularity [6]. This abstraction is done by means of an abstraction function that maps the symbols of the ground space (Σg) to the abstract space (Σa).

This function is obtained using an indistinguishability relation "~", defined as:

$(\forall x, y)\ x \sim y \equiv (\forall p \in \mathbf{R})(p(x) \equiv p(y))$

where **R** is the subset of the predicates of the theory which are relevant to the problem at hand.

Based on this relation, we can define a mapping function (k) that collapse an equivalence class in the ground space to a single constant symbol in the abstract space:

$(\forall x, y)\ x \sim y \equiv k(x) = k(y)$

In this way, we are making a process of *simplification* over the original ground space.

There is a problem associated to granularity and is that if the indistinguishability relation is not applicable over all predicates of the domain, there is always the possibility of an inconsistency, but this problem is inherent to almost any type of abstraction [3]. Because of this problem, we must choose very carefully the granularity we are going to work with; and to project some parts of the problem to a finer granularity to avoid the inconsistencies.

For the case in which we want to make an abstraction from the real numbers to the integer numbers, the indistinguishability relation will be, as Hobbs shows:

$(\forall x, y)\ x \sim y \equiv |x - y| < \varepsilon$

But in this case we have the problem that all the elements will finally be indistinguishable. To avoid this, we need a process of *idealization*, as Hobbs calls it, which consists in stipulating that certain elements are distinguishable. By means of this process, we state the frontiers between each class of equivalence and consequently, the exact form of the mapping function.

We will assume that each integer number t represents the interval of time $[t, t+1)$, so in most of the cases, the mapping function will have the form:

$$k_{\Sigma g \Sigma a}(x) = x \ div\ R_{\Sigma g \Sigma a} \qquad (3)$$

where $R_{\Sigma g \Sigma a}$ is the granularity factor and *div* is the integer division.

The process of abstraction can be applied several times, but now to the domain of integer numbers, so we obtain a hierarchy of abstractions, where each layer is an abstraction of the previous one. We base our representation of granularity in [2].

This convention has sense when we are working with different time granularities, where an hour, for example, contains 60 minutes, one minute, 60 seconds and so on. In the example of section 2, the solution following this approach is 5. This means that the tank becomes full during the 5th second. With this sense, the solution is exact, and if we want more precision, we can go down in the time hierarchy.

We will represent time points as a list of values corresponding to each granularity level, which will be ordered following the abstraction hierarchy.

Ex. 1 h, 10 m ≡ (1,10)

Using granularity we overcome (in a practical sense) the finiteness of integers. We can represent any number using the adequate granularity. Also we can represent temporal distances using the same representation.

As we have seen, one of the more important problems using integers is the representation of continuous change due to the discrete nature of integers, which makes it difficult to represent a continuous function. Taking the criterion that one integer represents the whole interval between itself and the next integer allows us to represent continuous functions as shown in figure 1.

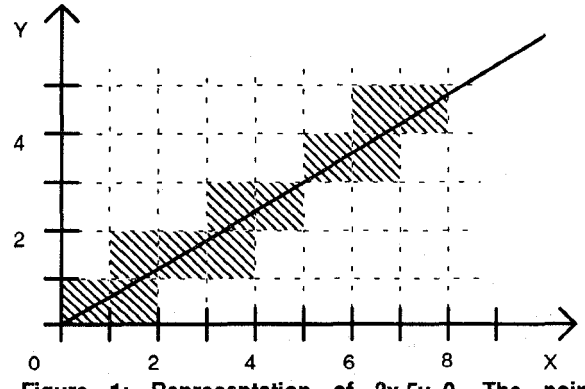


Figure 1: Representation of 3x-5y=0. The points belonging to the equation have been marked.

This representation is similar to the previous one we have presented if we take the error of the variables as a positive number between [0,1), but it must be noted that the theory behind is different.

Here we can also use the equation (1) for the representation of a linear equation changing the corresponding error, but now it is not an approximation but the exact representation according to granularity.

We obviously can also use the constraints (2) for a representation in a constraint solver.

$$C_1X_1 + \cdots + C_nX_n < C + C_1 + \cdots + C_n$$
$$C_1X_1 + \cdots + C_nX_n \geq C \qquad (4)$$

This approach also presents some problems inherent to granularity. The more important is that two points may be seen as one if they are very near due to the indistinguishability relation. This problem appears for example in the resolution of a system of linear equations, where we may obtain solutions that do not exist. Let's add the equation 3x+4y=33 to the previous one and let's try to find a solution, i.e. the intersection between the two lines (figure 2).

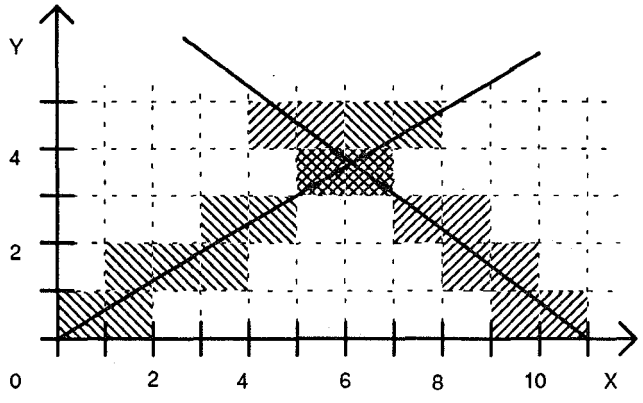


Figure 2: Representation of the system of equations.

The intersection of both lines gives the solution of the system, but there are two intersection points: the point [x=5, y=3] and the point [x=6, y=3]. At the present granularity of the problem both points follow the equations. To find the correct solution (as the solution is unique) we have to use a finer granularity and to abstract the solution later to the present granularity. At a finer granularity the problem still remain, but once we make the abstraction the solution will be one unless the real solution is near enough of both points, in which case we will be able to choose any one of them.

The solution of the system using real numbers is:

x = 6.11

y = 3.67

So the correct point should be [x=6, y=3]. We find also this result using a granularity of 2/10 with respect to the present granularity. We can see the result of this projection on figure 3.

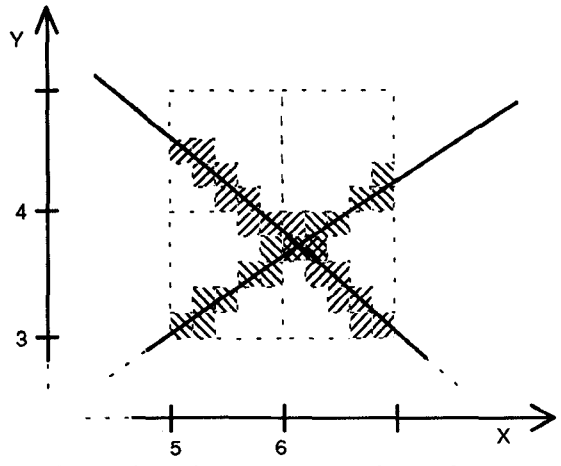
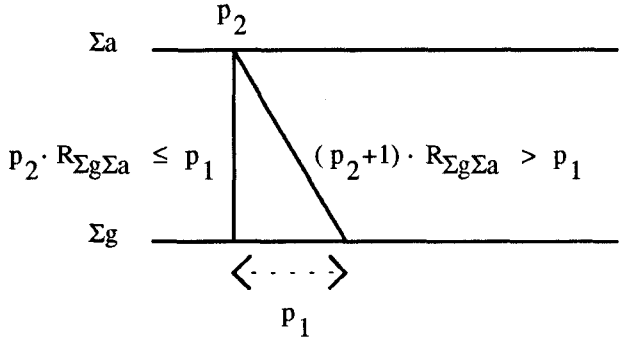


Figure 3: Projection to a granularity of 2/10.

The solution to the finer granularity is [x=(6,0), y=(3,3)] and [x=(6,1), y=(3,3)]. Making an abstraction we obtain [x=6, y=3], which is the expected solution.

This representation can be implemented in a constraint solver over finite domains. In equation (4) we have shown the representation for the linear functions. The abstraction function between different granularities can be represented in many cases by means of a linear constraint. The abstraction function shown in (3) is implemented by means of the constraints:

$$p_2 \cdot R_{\Sigma g \Sigma a} \leq p_1 < (p_2 + 1) \cdot R_{\Sigma g \Sigma a}$$



Where $p_1 \in \Sigma g$ and $p_2 \in \Sigma a$.

With this implementation, the projection or abstraction between different granularities are in fact the propagation through these constraints.

For this reason, it is desirable that we have control over the activation or deactivation of these constraints, so we have control over the process of abstraction and projection, to be able to avoid unnecessary or undesirable propagations.

The limitation of this implementation is that the constraint solvers are usually only for linear equations, so more complicated relations between granularities may give problems.

6.-Conclusions

We have presented in this paper two approaches to the representation of continuous change using discrete time, one using errors and the other using granularity, and we have seen that granularity has many advantages.

The approaches presented in the paper do not have a better representation of continuous change than real numbers, but they have the advantage of the use of integer arithmetic, which is simpler.

These representations have been thought as a possible basis for a reasoning mechanism over time and change using discrete time and implemented using a constraint solver over finite domains. Because of this reason, we have look for a simple mechanism ease to implement in a constraint solver. In any case, the approach presented here is not limited by the use of linear equations. It can be directly generalised to many types of equations found in different constraint solvers.

8.-Bibliography

[1] K. Van Belleghem, M. Denecker, D. De Schreye, "Representing continuous change in the abductive Event Calculus", Int. Conf. on Logic Programming, 1994.

[2] E. Ciapessoni, E. Corsetti, M. Miglioratti, E. Ratto, "Logical Specification of Real-Time Granular Systems in an Object Oriented Language", IJCAI 93, pp. 881-886, 1993.

[3] F. Giunchiglia, T. Walsh, "A theory of abstraction", Artificial Intelligence 57, pp. 323-389, 1992.

[4] P.J.Hayes, "Naive Physics 11:Ontology for Liquids", Formal Theories of the Commonsense World, Ablex, Norwood, N.J., 1984.

[5] P. Van Hentenryck, "Constraint Satisfaction in Logic Programming", MIT Press, 1989.

[6] J.R.Hobbs, "Granularity", IJCAI-85, pp. 432-435, 1985.

[7] R. Kowalski, M. Sergot, "A logic-based Calculus of Events", New Generation Computing, 4, 1986.

[8] J. M. Mc Carthy and P. J. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", Readings in Artificial Intelligence, Tioga, Palo Alto, Calif., pp 431-450, 1981.

[9] D.V.McDermott, "A temporal logic for reasoning about processes and plans", Cogn.Sci. 6 (1982), pp.101-155, 1982.

[10] R. Miller, M. Shanahan, "Reasoning about discontinuities in the Event Calculus", Proc. KR'96, Morgan Kauffman, 1996

[11] G. Nemhauser, L. Wolsey, "Integer and combinatorial optimization", John Wiley & Sons, 1988.

[12] J. A. Pinto, "Temporal reasoning in the Situation Calculus", PhD Thesis, University of Toronto, 1994.

[13] M. Shanahan, "Representing Continuous Change in the Event Calculus"; Proc. 9th ECAI, Stockholm, Sweden 1990.

[14] M. Vilain, H. Kautz, "Constraint propagation algorithms for temporal reasoning", AAAI-86, pp. 377-382, 1986.