Viability of Inductive Logic Programming as a learning mechanism in real-time systems

Maria do Carmo Nicoletti, Flávia O. S. de Sá Lisboa, Estevam Rafael Hruschka Jr
Computer Science Departament
UFSCar, Rod
Washington Luiz km 235
S. Carlos, SP
Brazil
{carmo@dc.ufscar.br, flavia@ifsc.usp.br, estevam@dc.ufscar.br}

ABSTRACT: *Automatic learning systems now consider the time as the significatnt component and it has good impact and hence it should be considered while developing these systems. Many studies have addressed this issue and one amont them is the Allen's temporal interval, based on a set of 13 relations that may hold between two time intervals. We in our study have kept the principal notion of identifying the several of temporal relations from data, using an inductive logic programming (ILP) system. This work addresses a series of automatic learning investigations. This study shows the following: The exploration of the impact of the negative training patterns on the induced relation, evidencing the necessary background knowledge for inducing the exact expression of the target concept and investigate the viability of ILP as a learning mechanism in real-time systems.*

Keywords: Temporal modelling, Temporal interval representation, Automatic learning of temporal relations, ILP, FOIL.

Received: 18 Decemeber 2010, Revised 21 January 2011, Accepted 27 Januaray 2011

1. Introduction

The representation of temporal knowledge aiming at implementing temporal reasoning is a recurrent problem in many knowledge areas, particularly in Artificial Intelligence (AI). The temporal aspect of situations and events is extremely relevant for implementing intelligent systems; the innumerous proposals to efficiently represent temporal information as well as surveys bound in the literature are evidence of the importance of this issue (e.g. [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]). As observed in [11], "[Temporal representation and reasoning] are becoming increasingly important with the advent of ubiquitous computing, grid computing and the Internet, where large amounts of information and processes are available, and where all these may be evolving in time". Vila in [12] mentions a few different areas of AI where representing time and implementing temporal reasoning are essential for the development of reliable and robust systems: (i) medical diagnosis and explanation – information about when, in what sequence the symptoms have occurred and how long they have persisted are critical in prescribing the correct treatment; (ii) planning – in the development of a plan, not only the duration of its actions and tasks should be carefully considered but also how they can be appropriately temporally ordered considering their many possible interaction over time; (iii) industrial process supervision – the control of an industrial process involves many temporal aspects such as past states, variable values evolution over time, the point of time particular sub-processes start (or end) etc. (iv) natural language understanding – mainly focusing on the verb tense in a sentence. Particularly when the focus is on natural anguage-based information systems, as pointed out in [13], "Applications such as information extraction, question-answering, summarization, visualization and developments in the Semantic Web can all benefit from analysis and interpretation along the temporal dimension.".

One of the most well-known and influential formalisms that deals with time representation is the Allen's temporal interval algebra [14] [15] [16] [17] [18] [19] [20] [21] [22] [23]. The main purpose of the work described in this paper is to explore how some of the temporal interval logic relations which have been stated as axioms, can be automatically learnt by a relational machine learning system. Of particular interest is to determine the influence of the negative examples on the induced concept. The FOIL (First Order Inductive Learner) system [24] [25], which uses Horn clauses as the language for representing and inducing knowledge, and one of the most successful ILP systems, was used in the experiments.

The paper is organized as follows: Section 2 briefly introduces the relational machine learning system FOIL in order to provide the basic knowledge for understanding how the system works and the information it requires. Section 3 presents some of the main concepts and the basic relations related to Allen's temporal intervals. Particularly, the learning of seven of Allen's 13 basic relations will be main goal to achieve when using the relational learning system FOIL, as described in Section 4. Finally in Section 5 comments about the work and a few directions into future work are presented.

2. The FOIL System – Main Aspects

As stated in [26], "Inductive Logic Programming (ILP) is the area of AI which deals with the induction of hypothesized predicate definitions from examples and background knowledge. Logic programs are used as a single representation for examples, background knowledge and hypotheses. ILP is differentiated from most other forms of Machine Learning (ML) both by its use of an expressive representation language and its ability to make use of logically encoded background knowledge." FOIL is an ILP system which induces function-free Horn clause representations of relations. The inductive process implemented by FOIL is based on examples (positive and negative examples of the relation to be learnt) and a given background knowledge (BK), known as domain theory (a set of presumably known relations which are fed to the system in their extensional form, i.e., as ground facts). FOIL learns a description of the target relation represented in terms of the target and, eventually, of the other relations given as BK. So, given the positive and negative examples that define the relation to be learnt and the positive examples that define each of the relations that are part of the BK, FOIL's task is to learn a function-free Horn clause definition of the target relation.

"FOIL's definitions consist of ordered clauses. These clauses are extended forms of function-free Horn clauses in that the body literals may be negated, where the meaning of negation accords with the Prolog convention. There is an option to exclude such negated literals if the user wishes. The head of a clause is always a literal consisting of the target relation, with unique variables for each argument position. The literals in the body may be often forms (including negations), subject to various restrictions. Every literal must contain at least one variable bound by the head of the clause or a preceding body literal; such variables will be referred to as 'previously bound'. FOIL checks whether discrete types have constants in common and if so considers variables of such compatible types to satisfy type restrictions on relations" [25].

When learning a definition, FOIL uses a covering approach (an example is covered by a clause that proves it). FOIL repeatedly learns a clause and then removes the positive examples that are covered by it from the set of positive examples to be covered. Once the definition is complete (all the positive examples are covered), FOIL prunes out redundant clauses. The whole process is much more elaborated than the brief view given above. A detailed and in-depth description of FOIL can be found in [24], [25] [27]. As an example of how to use FOIL for learning a relation, consider the graph given in Figure 1. Suppose the relation *path(A,B)* (representing that there is a path between two nodes A and B of the graph) is the target relation whose general representation should be learnt by FOIL. Suppose also that examples of relations are known, such as (i) *arc(A,B)* (expressing that there is an arc from node A to B), (ii) *node(A)*, expressing that A is a node in the graph.

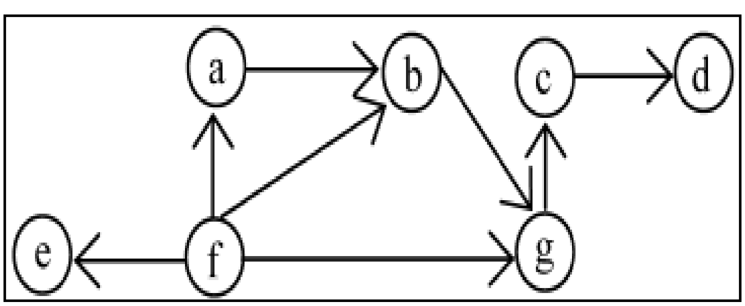

Figure 1. A graph defined by 7 nodes and 8 arcs

Input data to FOIL shown in Figure 2 follows the required syntax where the header of a relation, other than the target relation, begins with an ‘*’ and consists of the relation name, argument types and optional keys, followed by a sequence of lines containing examples. Each example consists of constants separated by commas (argument separator) and must appear in a single line. Due to space restriction, however, the convention of placing one example per line will not be followed in this paper. A line with the character ‘.’ indicates the end of the extensional description of the corresponding relation. Note in Figure 2 that the negative examples of relation *path/2* have not been given – if all the positive examples are known, the negative examples can be generated under the CWA (Closed World Assumption). When referring to predicates, their arity (i.e., number of arguments) can be omitted for the sake of simplicity, when that does not introduce ambiguity.

X: a,b,c,d,e,f,g.	a,g	f,c	*arc(X,X)	f,e	b
path(X,X)	b,c	f,d	a,b	f,g	v
a,b	b,d	f,e	b,g	c,d	d
a,c	b,g	f,g	g,c	.	e
a,d	c,d	g,c	f,a	*node(X)	f
	f,a	g,d	f,b	a	g
	f,b				

Figure 2. Input for FOIL to learn relation *path/2* given two relations as BK. Due to space restriction the convention of having one example per line has been abolished. The information displayed in six columns above should be read from left to right, in sequence

Considering the input file as described in Figure 2, FOIL induces the two clauses below which are a general description that can be used to find a path in any graph. Note that although the unary relation *node/1* has been given to FOIL as background knowledge, its contribution for representing the concept of path (in the given context) is not relevant enough and so, *node/1* is not part of the set of clauses induced by FOIL.

path(A,B) :- arc(A,B).

path(A,B) :- arc(A,C), path(C,B).

3. A Brief Overview of Allen’s Temporal Interval Relations

The formalism known as Temporal Interval Relations (TIR) (or Allen’s Interval Algebra), based on binary interval relations to represent and reason about temporal knowledge, was initially proposed by Allen in [14], [15]. Since then it has been the subject of several research works particularly related to its use in different domains and possible extensions it allows [2] [28] [29].

The information about TIR given next has been compiled from the many sources that discuss this formalism, particularly from [21]. TIR has a simple linear model of time, having one primitive object referred to as *time period* and one primitive relation named *meets/2*. A time period can be thought of as the time associated with some event occurring or some property holding in the world. Although the intuitive idea associated with the *meets/2* relation is very clear, its description as a formal concept (aiming at its implementation) is not trivial, since it involves a previous definition of the adopted granularity for time as well as the adopted representation for time interval.

Allen in [15] discusses the appropriateness of using time points as opposed to time intervals, as the basis for temporal representation and reasoning. As Mani and coworkers comment in [13], “Depending on the representation and the choice of primitive (intervals or instants, or both), a variety of different temporal relations between times can be defined.”

In spite of TIR taking temporal intervals as primitive, the formalism contemplates the possibility of, assuming a model consisting of a fully ordered set of points of time, representing an interval T as an ordered pair of points of time $< t-, t+ >$, satisfying the pre-condition of the first point being lesser than the second, i.e., $t- < t+$. For the purposes of this paper this was the representation adopted, noted by [t-, t+], in spite of some inconveniences it can bring, which have been vaguely described by Allen in [15] as “being too uniform and does not facilitate structuring the knowledge in a way which is convenient for typical temporal reasoning task”. Vila in [12] presents a detailed discussion related to time representations (time points, intervals); Allen and Hayes in [17] and [18] consider extending TIR introducing the concept of time point.

Table 1**.** Allen’s 5 basic binary temporal interval relations and their equivalent relations assuming intervals (time periods) are defined by time points, i.e., A = [a”,a+] (a” < a+) and B= [b”, b+] (b” < b+), where {a”,a+, b”,b+} †” “!. In the second column the default conditions (derived from the fact that for interval T, t” < t+ always holds) are also presented. The condition(s) in bold face are the ones that characterize the corresponding relation. Comma represents the logical ‘and’ and semi-colon the logical inclusive ‘or’.

Table 1. presents Allen’s 5 basic binary temporal relations between intervals, namely *meets*, *before*, *overlaps*, *during*, and *equal*.

Interval Relation	Conditions of interval endpoints	Pictorial representation
meets(A,B)	a” < b”, a” < b+	AAAA
	a+ = b”, a+ < b+	BBBBBBB
before(A,B)	a” < b”, a” < b+	AAAA BBBBBBB
	a+ < **b**”, a+ < b+	
overlaps(A,B)	**a**” < **b**”, a” < b+	AAAA
	a+ > **b**”, **a+** < **b+**	BBBBBBB
during(A,B)	(**a**” > **b**”, **a+** =< **b+**) ;	
	(**a**” >= **b**”, **a+** < **b+**)	See Table 2
equal(A,B)	**a**” = **b**”, a” < b+	AAAAA
	a+ > b”, **a+ = b+**	BBBBB

The second column of Table 1 shows the equivalent relations on endpoints; the four conditions on time points are presented (although some of them can be logically entitled by others). According to Allen, the subdivision of the *during* relation into three others namely *starts*, *finishes* and a new *during*, provides a better computational model. The three new temporal interval relations are shown in Table 2. Considering the subdivision of *during* relation the number of basic temporal interval relations adds up to 7.

Reminding that if R is a binary relation, the converse (reversed) relation of R, written R”1, is a relation such that yR”1x if and only if xRy. The 7 basic relations shown in Table 1 and Table 2 can be expanded into 13, if their reversed relations are considered (the reversed relation of *equal* is itself). Table 3 names the reversed relations associated to 6 basic temporal interval relations (*equal* excluded).

4. Learning Temporal Interval Relations with FOIL – the Basic Learning Cases

This section describes the necessary information (positive examples and BK) for using FOIL to learn each of Allen’s 7 basic temporal relations shown in Table 1 and Table 2. For the experiments it was used the version 5.1 of FOIL downloaded from http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/learning/systems/foil. The FOIL system runs on a Virtual Machine with Linux Ubuntu version 10 with gcc compiler.

For each relation a table containing the positive examples (PE), the background knowledge (BK) and the induced concept (IC) is presented. To ease the understanding of the information in each of the following tables a figure (contained in Figure 3) explicitly representing the given knowledge is presented. There was also an attempt to minimize the number of examples of the relation to be learnt as well as of the BK given to FOIL as input. The motivation for that was to determine the least possible amount of information to give to FOIL that would still allow the system to induce the proper expression of the relation, as

Interval Relation	Conditions of interval endpoints	Pictorial representation
starts(A,B)	**a**” = **b**”, a” < b+	AAAA
	a+ > b”, **a+** < **b+**	BBBBBBB
finishes(A,B)	**a**” > **b**”, a” < b+	AAAA
	a+ > b”, **a+ = b+**	BBBBBBB
during(A,B)	**a**” > **b**”, a” < b+	AAAA
	a+ > b”, **a+** < **b+**	BBBBBBB

Table 2. Subdividing the *during* relation from Table 1 into three new relations: *starts*, *finishes* and a new *during*. Notation and convention follow those established for Table 1.

Interval Relation	Reversed Relation	Equivalences
meets	met_by	meets(A,B) a” met_by(B,A)
before	after	before(A,B) a” after(B,A)
overlaps	overlapped_by	overlaps(A,B) a” overlapped_by(B,A)
starts	started_by	starts(A,B) a” started_by(B,A)
finishes	finished_by	finishes(A,B) a” finished_by(B,A)
during	contains	during(A,B) a” contains(B,A)

Table 3. Temporal basic interval relations, their correspondent reversed relations and the equivalence relation between each pair of them.

described in Table 1. As mentioned before, in the following tables the knowledge will not be presented according to the syntax required by FOIL nor the combinations of different BK and sets of positive examples will be exhaustively covered. The main goal is to show how the basic temporal relations can be learnt and how the background knowledge and positive examples can influence the induced expression. In the experiments a time interval is represented by its two endpoints. As a consequence, Allen's binary relations are represented by four arguments - the first two define the first temporal interval and the last two, the second temporal interval. It is assumed that each of these pair of values defines a proper temporal interval, as discussed in Section 3, i.e., a valid temporal interval T is defined as an ordered pair of points of time <t-, t+>, satisfying the pre-condition of the first point being lesser than the second, i.e., t- < t+). For the experiments described in the following subsections, the pictorial examples of the respective relation, as shows Figure 2, have been used.

In the following tables: (a) LE identifies a number associated with a learning experiment; (b) due to space restrictions, the line with the character "." indicating the end of a listing of examples will be suppressed; (c) column PE lists the positive examples of the relation to be learnt; (d) column NE lists the negative examples – if none is provided by the user, FOIL automatically generates them under the CWA. When the CWA is used, the column informs the number of negative examples generated by FOIL; (e) the presence of *** in column IC means that FOIL was unable to induce an expression for the concepts represented by the positive examples, given the KB and NE examples shown in the corresponding line of the table; (f) FOIL requires the definition of the type of data the examples represent. For all the experiments, this request has been accomplished by having, at the beginning of the input file, the type definition: X:1,2,3,4 or X:1,2,3,4,5 or X:1,2,3,4,5,6 or X:1,2,3,4,5,6,7, depending on the relation to be learnt (see Figure 3).

4.1 Learning the *meets/4* relation

Figure 3(a) pictorially shows the three positive examples of relation *meets/4* initially given to FOIL. Table 4 shows the learning results obtained by FOIL in three learning environments: #1: 3 positive examples, no negative examples and no BK; #2: 3 positive examples, 1 negative example and no BK; and #3: 2 positive examples, 1 negative example and no BK.

In #1 no NEs were given and FOIL used the CWA to automatically generate 253 negative examples (all possible combinations with repeated elements of numbers 1,2,3,4, except for the three given as positive examples). Many of the negative examples

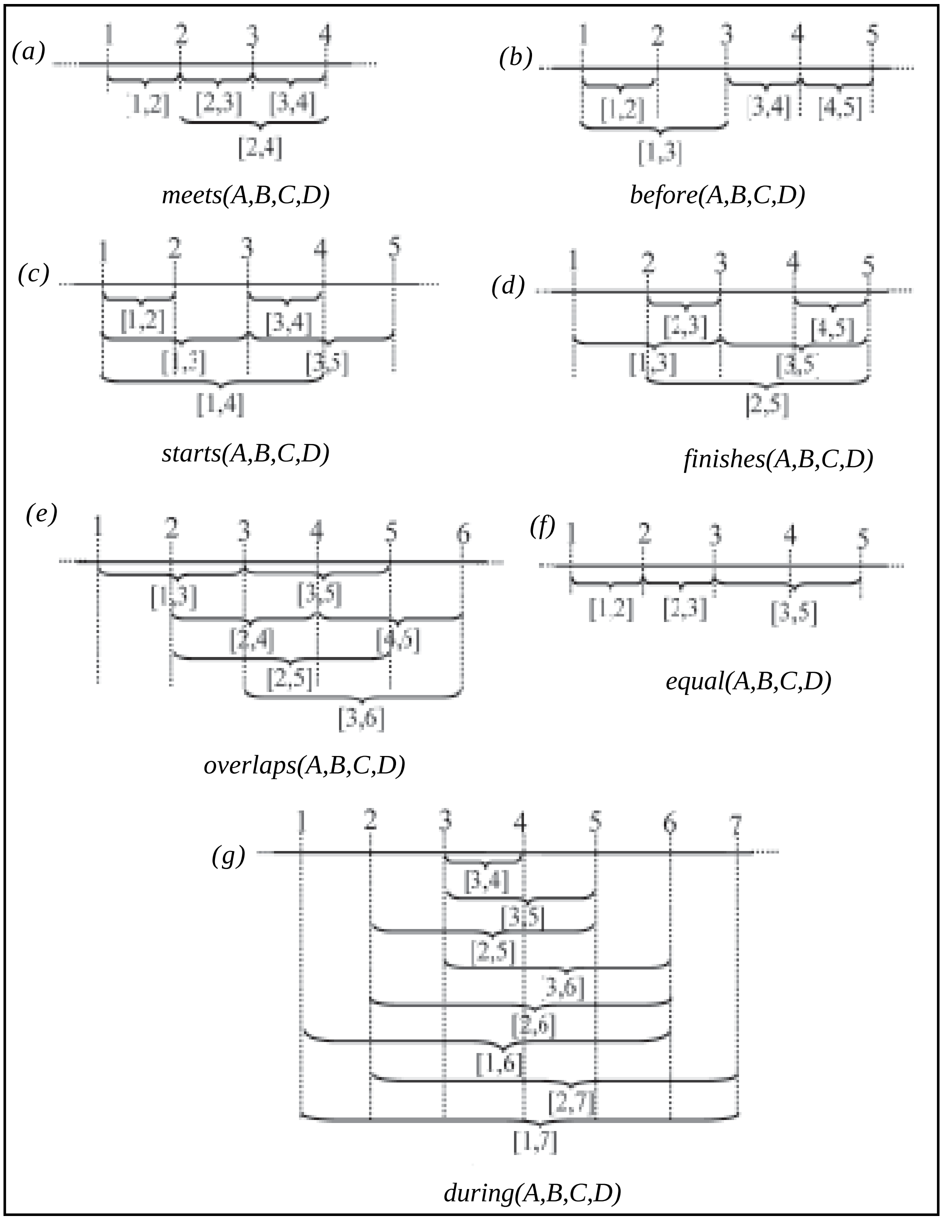

Figure 3. Pictorial examples used for learning the 7 basic temporal relations

were actually positive examples (e.g. 1,3,3,4 is considered a negative example under CWA, given the three positive examples). In #2 and #3 FOIL induced the condition that should be satisfied by the upper limit and lower limit of the first and the second temporal intervals respectively (i.e., B=C). Particularly #3 shows that it is possible to learn the concept with only 2 well-defined Positive examples and 2 welldefined negative examples. For the learning of the meets/4 relation no BK was required.

4.2 Learning the before/4 relation

Figure 3(b) pictorially shows the situations that inspired the positive examples of relation before/4 initially given to FOIL. Table 5 shows the learning results obtained by FOIL in four learning environments described as: #1: 3 positive examples, no negative examples and no BK; #2: 3 positive examples, no negative examples and one relation as BK; #3: 3 positive examples, 2 negative examples and one relation as BK and finally #4: three positive examples, 3 negative examples and one relation as BK.

Similarly to what happened with meets/4, the use of CWA by FOIL (since negative examples have not been given) generates false negatives - FOIL tries to generalize based on wrong information and does not succeed (see experiment #1). Considering that 3 positive examples were given, by the CWA the examples 1,2,3,5 and 2,3,4,5, for instance, are considered negative examples when they clearly are positive examples. In experiment #2 the CWA was maintained and the relation lessthan/2 was given as BK. Although FOIL succeeds to induce an expression for relation before/4, that definitely is not a good representation of the concept, since it rules out many positive examples that, due to the use of the CWA, have been considered as negative examples. The input to FOIL in #3 and #4 differs only in relation to an extra negative example given to the system in #4 which, definitely, is critical for the system to induce the expected representation of the concept.

LE	PE (meets(X,X,X,X))	NE	BK	IC
#1	1,2,2,3 2,3,3,4 1,2,2,4	CWA (#NE: 253)	∅	***
#2	1,2,2,3 2,3,3,4 1,2,2,4	1,2,3,4	∅	meets(A,B,C,D) :- B=C.
#3	1,2,2,3 1,2,2,4	1,2,3,4 2,3,1,2	∅	meets(A,B,C,D) :- B=C.

Table 4. Learning meets/4 based on Figure 3(a) where X: 1,2,3,4 (for FOIL)

LE	PE (before(X,X,X,X))	NE	BK	IC
#1	1,2,3,4 1,3,4,5 1,2,4,5	CWA (#NE: 622)	∅	***
#2	1,2,3,4 1,3,4,5 1,2,4,5	CWA (#NE: 622)	lessthan(X,X) 1,2 2,4 1,3 2,5 1,4 3,4 1,5 3,5 2,3 4,5	before(A,B,C,D) :- ¬lessthan(E,A), lessthan(B,C), lessthan(C,D), lessthan(E,B), ¬before(A,C,F,D).
#3	1,2,3,4 1,3,4,5 1,2,4,5	3,4,1,2 2,3,3,4	Same as #2	***
#4	1,2,3,4 1,3,4,5 1,2,4,5	3,4,1,2 1,3,2,5 2,3,3,4	Same as #2	before(A,B,C,D) :- lessthan(B,C).

Table 5. Learning *before/4* based on Figure 3(b) where X: 1,2,3,4,5 (for FOIL)

4.3 Learning the *equal/4* relation

The learning of relation *equal/4* was based on the situation displayed in Figure 3(f). The information given to FOIL in the experiments is described in Table 6. The four experiments were conducted with the same positive examples and no BK. They vary in relation to the negative examples given. In #1: no negative example; #2: 1 negative example; #3: 1 negative example and #4: 2 negative examples (those that were individually given in #2 and #3).

In #1 the use of CWA provoked many positive examples to be considered as negative (e.g. 1,3,1,3) and FOIL ended without learning any concept. Note that in #2 and #3 only one suitable negative example was given and only a partial definition of the concept was induced. However in #4, when the two negative examples are given, FOIL was provided with the necessary information to induce the conjunction of both conditions that represents the correct expression of *equal/4*.

4.4 Learning the *starts/4* relation

The information pictorially shown in Figure 3(c) has been used for learning the relation *starts/4* in four attempts, as escribed in Table 7. In experiment #1 the CWA was adopted by FOIL since no negative example was provided. Similarly to what happened in the three previous relations, many false negatives were created and FOIL failed to induce an expression for the concept. Experiments #2, #3 and #4 were conducted with the same set of positive examples and with the same relation given as BK. They vary in relation to the negative examples provided. The two negative examples given in #3 were unable to provide FOIL with sufficient information to induce a suitable expression of the concept; only when a few more negative examples are provided (#4) FOIL induces the expected representation for *starts/4*.

LE	PE (equal(X,X,X,X))	NE	BK	IC
#1	1,2,1,2 3,5,3,5 2,3,2,3	CWA (#NE: 622)	∅	***
#2	Same as #1	1,3,2,3	∅	equal(A,B,C,D) :- A=C.
#3	Same as #1	1,2,1,4	∅	equal(A,B,C,D) :- B=D.
#4	Same as #1	1,2,1,4 1,3,2,3	∅	equal(A,B,C,D) :- A=C, B=D.

Table 6. Learning *equal/4* based on Figure 3(f) where X: 1,2,3,4,5 (for FOIL)

LE	PE (starts(X,X,X,X))	NE	BK	IC
#1	1,2,1,3 1,3,1,4 1,2,1,4 3,4,3,5	CWA (#NE: 621)	∅	***
#2	Same as #1	2,5,5,6	lessthan(X,X) 1,2 2,3 3,5 1,3 2,4 3,6 1,4 2,5 4,5 1,5 2,6 4,6 1,6 3,4 5,6	starts(A,B,C,D) :- A=C.
#3	Same as #1	2,5,5,6 2,5,2,3	Same as #2	starts(A,B,C,D) :- ¬lessthan(E,C).
#4	Same as #1	2,5,5,6 3,6,2,5 2,5,2,3 3,6,4,5 3,6,4,6 3,6,1,2 3,6,2,6	Same as #2	starts(A,B,C,D) :- A=C, lessthan(B,D).

Table 7. Learning starts/4 based on Figure 3(c) where X: 1,2,3,4,5,6 (for FOIL)

4.5 Learning the *finishes/4* relation

The *finishes/4* relation defines when two time intervals are such that the first starts after the second has started and both end at the same ending point. Figure 3(d) shows a diagram of the situation used as information to FOIL, to conduct the 7 learning experiments detailed in Table 8, all using the same set of four positive examples. The CWA was used by FOIL in #1 and, as expected, due to the many false negatives generated, FOIL failed to induce an expression for the concept. As the only negative example given in #2 was 2,5,2,3 when generalizing FOIL adds the literal A<>C in the body of the clause, since A=C in the negative example given. The two negative examples given in #3 direct FOIL to include literal *lessthan/2* in the induced expression. Note in #4 that given the five negative examples (the two from #3 plus three new ones), FOIL induces the same expression as in #3. Increasing the number of NE to 7, as in #5, FOIL includes a negative literal in the induced expression. If FOIL is instructed *via* command prompt, as in #6, to ignore negative literals, the expected expression for *finished/4* is obtained. The same expression, however, can be induced without instructing FOIL, by including a carefully chosen negative example, as done in #7.

* obtained with the –n argument when running FOIL. The parameter instructs FOIL not to consider negative literals.

4.6 Learning the *overlaps/4* relation

As can be seen in Table 1, the definition of *overlaps/4* requires three conditions on interval endpoints; a careful analysis of the appropriate examples that can direct FOIL into the direction of inducing the proper definition should be conducted. As discussed before, the *overlaps* relation is true when the first temporal interval overlaps the second. Accordingly to the semantics associated with Allen's *overlaps* relation, the relation is true if (a) the first temporal interval starts earlier than the second, (b) the second starts while the first is still going on and (c) the first interval ends while the second is still going on.

LE	PE(finishes(X,X,X,X))	NE	BK	IC
#1	2,3,1,3 4,5,2,5 4,5,3,5 3,5,2,5	CWA (#NE: 1292)		***
#2	Same as #1	2,5,2,3	Same as #2(Table 7)	finishes(A,B,C,D) :-A<>C.
#3	Same as #1	2,5,2,3 2,5,3,5	Same as #2 (Table 7)	finishes(A,B,C,D) :- lessthan(C,A).
#4	Same as #1	2,5,2,3 2,5,3,4 2,5,2,6 2,5,5,6 2,5,3,5	Same as #2 (Table 7)	finishes(A,B,C,D) :- lessthan(C,A).
#5	Same as #1	2,5,2,3 2,5,5,6 2,5,2,6 3,4,4,6 2,5,3,5 3,4,1,2 2,5,3,4	Same as #2 (Table 7)	finishes(A,B,C,D) :- B=D, ¬lessthan(A,C).
#6	Same as #1	Same as #5	Same as #2 (Table 7)	(*) finishes(A,B,C,D) :- B=D, lessthan(C,A).
#7	Same as #1	2,5,2,3 2,5,5,6 2,5,2,6 3,4,4,6 2,5,3,5 3,4,1,2 2,5,3,4 3,4,3,4	Same as #2 (Table 7)	finishes(A,B,C,D) :-B=D, lessthan(C,A).

Table 8. Learning finishes/4 based on Figure 3(d) where X: 1,2,3,4,5,6 (for FOIL)

Figure 3(e) is a diagram of the situations given as examples to FOIL in the eight experiments detailed in Table 9. Experiments #2 to #7 kept constant the number of positive examples and increased the number of negative examples starting with 1 (#2), followed by 2 (#3), 5 (#4), 7 (#5) and 12 (#6), when FOIL finally gathered enough information to induce the expected representation of *overlaps/4*, i.e., a conjunction of three conditions on time points of both intervals, as shown in Table 1.

The following experiments in Table 9 (i.e. #7 and #8) tried to reduce the number of positive examples used, maintaining the 12 negative examples. Experiment #7 differs from #6 in relation to the number of positive examples, which was decreased by one. FOIL still managed to induce the approp riate expression for the concept. However,when the number of positive examples dropped to 4, FOIL failed to find a representation for the concept consistent with the given examples and BK.

4.7 Learning the *during/4* relation.

Figure 3(g) presents a diagram of the situations given as examples to FOIL as learning situations for relation *during/4.* Due to space restriction Table 10 only presents 5 out of the 7 conducted experiments. Also, the examples of *lessthan/2* given as BK are those already presented in Table 7, including the following pairs: 1,7 2,7 3,7 4,7 5,7 and 6,7 since X: 1,2,3,4,5,6,7.

The experiment where no negative examples were given, no BK was given and #PE=13, resulted in a set of 2388 automatically created negative examples (CWA) with FOIL ending without inducing any expression. The same strategy used before was adopted – to fix the number of positive examples and increase (per experiment) the number of negative examples. In all the experiments, except for the one described above, BK was represented by only one relation (*lessthan/2*) with 21 examples.

As can be seen in Table 10, FOIL was able to induce the expected concept in experiment #2, based on #PE=13 and #NE=10 and #NE=10 (and on the BK *lessthan/2*). In the following experiment (#3) the number of negative examples was reduced to 6 and FOIL only induced a partial expression of the concept. In Experiment #4 the negative examples are back to 10 and a reduction in the number of positive examples was tried (#PE=3) – FOIL succeeds in inducing the correct expression for the concept. The following experiment tries again a reduction in the number of negative, maintaining the reduced number of positive as the previous experiments and FOIL ends without inducing any expression.

LE	PE(overlaps(X,X,X,X))	NE	BK	IC
#1	1,3,2,4 2,4,3,6 1,3,2,5 2,5,3,6 3,5,4,6 2,5,4,6	CWA (#NE: 1290)	∅	***
#2	Same as #1	3,6,4,6	Same as # (Table 7)	overlaps(A,B,C,D):- B<>D.
#3	Same as #1	3,6,4,6 3,6,2,6	Same as #2 (Table 7)	overlaps(A,B,C,D):- B<>D.
#4	Same as #1	3,6,4,6 3,6,4,5 3,6,2,6 3,6,1,2 3,6,2,5	Same as #2 (Table 7)	overlaps(A,B,C,D):- lessthan(B,D).
#5	Same as #1	3,6,4,6 3,6,1,2 3,6,2,6 1,2,2,4 3,6,2,5 1,2,3,5 3,6,4,5	Same as #2 (Table 7)	overlaps(A,B,C,D):- lessthan(B,D), lessthan(C,B).
#6	Same as #1	3,6,4,6 1,2,3,5 3,6,2,5 3,5,3,6 3,6,4,5 3,5,2,4 3,6,1,2 3,5,2,6 1,2,2,4 3,5,1,2 3,6,2,6 3,5,3,4	Same as #2 (Table 7)	overlaps(A,B,C,D):- lessthan(A,C), lessthan(B,D), lessthan(C,B).
#7	1,3,2,4 2,4,3,6 1 ,3,2,5 2,5,3,6 3,5,4,6	Same as #7	Same as #2 (Table 7)	overlaps(A,B,C,D):- lessthan(A,C), lessthan(B,D), lessthan(C,B).
#8	1,3,2,4 3,5,4,6 1,3,2,5 2,4,3,6	Same as #7	Same as #2 (Table 7)	***

Table 9. Learning overlaps/4 based on Figure 3(e) where X: 1,2,3,4,5,6 (for FOIL)

5. Conclusions and Further Work

Allen's temporal interval logic goes far beyond the basic issues discussed in this paper since it is the basis for representing events and actions. As briefly mentioned in Section 3, Allen's temporal theory is based on a primitive object, the time period, and one primitive binary relation, the *meets* relation. Next, the theory introduces an axiomatization of the *meets* relation, by establishing 5 axioms based on the following intuitive notions: (1) every period has a period that meets it and another that it meets; (2) time periods can be composed producing a larger time period; (3) periods uniquely define an equivalence class of periods that meet them (4) these equivalence classes also uniquely define the periods and finally (5) periods can be ordered. Based on both primitives i.e., time period and the *meets/2* relation, and considering the five axioms, several intuitive relationships that could hold between time periods could be defined, such as those already seen in tables 1, 2 and 3.

The work described in this paper will continue by (i) investigating the original time period representation proposed by Allen [14][30] i.e., as a primitive concept and not as a concept represented by time points as adopted in this paper; (ii) running automatic learning experiments with FOIL to evaluate the viability of having time period as primitive; (iii) comparatively evaluate results obtained with both temporal period representations taking into account the necessary examples and background knowledge given to the system for promoting the induction of a consistent expression of the target relation to be learnt.

As it is well established in the literature and very obvious from all the experiments described in this paper, the examples given to the learning system play a crucial role and are determinant for inducing a 'good' representation of the concept. Particularly when using ILP systems, the relevance of the examples and BK seems even more crucial. As can be seen in many of the experiments described in this paper, the use of the closed world assumption can be misleading and consequently, the 'right' choice of negative examples plays a fundamental role in inducing a sound and representative expression of the concept. Choosing a suitable set of negative examples, however, can be a particularly 'tricky' task which, most certainly, will not be feasible to be automatically conducted in a real time environment.

LE	PE(during(X,X,X,X))	NE	IC
#1	3,4,2,5 3,4,1,7 3,5,1,7 3,4,2,6 3,5,2,6 2,5,1,6 3,4,1,6 3,5,1,6 2,5,1,7 3,4,2,7 3,5,2,7 3,6,2,7 3,6,1,7	2,5,2,3 2,5,2,6 2,5,3,6 2,5,3,4 2,5,6,7	during(A,B,C,D):- lessthan(C,A).
#2	3,4,2,5 3,4,1,7 3,5,1,7 3,4,2,6 3,5,2,6 2,5,1,6 3,4,1,6 3,5,1,6 2,5,1,7 3,4,2,7 3,5,2,7 3,6,2,7 3,6,1,7	2,5,2,3 2,5,6,7 3,6,4,5 2,5,2,6 3,6,4,6 3,6,1,2 2,5,3,6 3,6,2,6 2,5,3,4 3,6,2,5	during(A,B,C,D):- lessthan(B,D), lessthan(C,A).
#3	3,4,2,5 3,4,1,7 3,5,1,7 3,4,2,6 3,5,2,6 2,5,1,6 3,4,1,6 3,5,1,6 2,5,1,7 3,4,2,7 3,5,2,7 3,6,2,7 3,6,1,7	3,6,4,6 3,6,2,5 3,6,1,2 3,6,2,6 3,6,4,5	during(A,B,C,D):- lessthan(B,D).
#4	3,4,2,5 3,4,2,6 3,4,1,6	2,5,2,3 2,5,6,7 3,6,4,5 2,5,2,6 3,6,4,6 3,6,1,2 2,5,3,6 3,6,2,6 2,5,3,4 3,6,2,5	during(A,B,C,D):- lessthan(B,D), lessthan(C,A)
#5	3,4,2,5 3,4,2,6 3,4,1,6	2,5,2,3 3,6,2,5 2,5,2,6 3,6,4,5 2,5,3,6 3,6,1,2	***

Table 10. Learning during/4 based on Figure 3(g) where X: 1,2,3,4,5,6,7 (for FOIL)

Acknowledgements

The authors thank the IFSC/USP (São Carlos) and CNPq for their financial support.

References

[1] Bellini, P., Mattolini, R., Nesi, P. (2000) .Temporal logics for real-time system specification, *ACM Computing Surveys*, 32 (1) 12–42.

[2] Schockaert, S, De Cock, M, Kerre, E. E.(2008). Fuzzifying Allen's temporal interval relations, *IEEE Trans. on Fuzzy Systems*, 16 (2) 517–533.

[3] Furia, C., Mandrioli, D., Morzenti, A., Rossi, M., (2010). Modeling time in computing: a taxonomy and a comparative survey, *ACM Computing Surveys*, 42 (2) Article 6.

[4] Knight, B., Ma, J. (1994). Time representation : a taxonomy of temporal models. Artificial Inteligence Review, N. 7, 401–419.

[5] Long, D (1989). A review of temporal logics. The Knowledge Engineering Review, 4 (2) 141–162.

[6] Chitaro, L., Montanari, A. (2000). Temporal representation and reasoning in artificial intelligence: Issues and approaches. *Annals of Mathematics and Artificial Intelligence*, 28, 47–106

[7] Bettini, C. (1997). Time dependent concepts: Representation and reasoning using temporal description logics, *Data & Knowledge Engineering*, 22 (1) 1–38.

[8] Ma, J., Knight, B. (1996). A general temporal theory, *The Computer Journal*, 37 (2) 114– 123.