Creating Temporal Categories for an Ontology of Time

Joachim De Beule

VUB, ARTI-lab,
joachim@arti.vub.ac.be

Abstract

A mechanism is described that enables a robotic agent to create temporal categories for conceptualizing the world. The creation of a new category is triggered when the agent is unable to temporally distinguish an event from the other events in the context using already adopted categories. This is different from most other approaches where ontological categories are defined by humans and the ontologies are fixed in advance.

1 Introduction

In this paper the problem of how a robotic agent can create and maintain an ontology in the domain of time is addressed. By an ontology we mean a set of categories or distinctions with which he can conceptualize the world. Our agents need such a set of temporal distinctions because they have to communicate temporal information.

There is a still growing interest in ontology research. For example agents in a multi-agent setting need a shared ontology to cooperate [11]. Natural language processing systems need an ontology to successfully determine the contents of a phrase or request.

But in contrast with what is proposed in this paper, most ontologies consist of a fixed set of definitions and relations reflecting the domain knowledge of its designers. In the CYC project [5] common sense knowledge is obtained from humans and collected in a knowledge base. For the semantic web (see e.g. [10]) numerous ontologies are being designed and standardized. In the domain of time, Allen, Comrie and Reichenbach [1, 3, 6] used logic and analyzed natural languages to determine the semantics of grammatical tense categories. All these examples have in common that an ontology is constructed by hand and that it is fixed once and for all.

Such an approach has several disadvantages [9]. For example, the way in which a domain is structured and conceptualized *by humans* is not fixed but differs from culture to culture. This is especially true for the semantics of linguistic constructions since language is a conventional system, and the meaning of a grammatical tense category is conventionally determined [4]. Different languages not only express temporal information differently, the information is also conceptualized

differently. Some tense systems require a differentiation between the recent and the distant past while others do not have an obligatory tense system at all. In addition, the conceptualization and verbalization schemes of natural language are subject to constant evolution. Not only because the world changes (e.g. as technology advances more precise temporal distinctions are needed) but also because there is no universal way to conceptualize the temporal properties of the world. As new language users enter the community, new distinctions get introduced or existing temporal constructions are reinterpreted etc. If the robotic agents have to communicate with humans they should be adequately equipped to cope with this.

Thus, we claim that a conceptualization scheme or ontology can not exist independently from the environment in which it is used. And for an agent of a language community, the environment includes the set of conventions defining the communication system of the community. If two agents want to communicate about something they have to agree upon both *what* to express about it (how to conceptualize it) and *how* to express it (how to verbalize the conceptualization.) As such, the emergence of a set of categories shared between the agents requires an ongoing negotiation and a co-evolutionary coupling between the categories that make up an agent's ontology and the forms to express them [9]. In other words, the agents themselves should decide whether their ontology of time is adequate or not and, if needed, adjust it to conform to the consensus in the population. The measure for adequacy is not only whether the agent is capable of conceptualizing the temporal information it wants to express but also whether it is successful in communicating these conceptualizations.

To summarize, the ultimate goal is to create a population of artificial autonomous agents establishing a set of conventions to communicate temporal information. For this to be possible, the agents have to agree upon an ontology in the domain of time. Thus, an agent should be capable of doing at least three things. First, he should be capable of creating new categories. Second, he should be capable of incorporating these categories into his language repertoire. Third, he should be capable of adapting his language and ontology in order to conform to the (emerging) consensus in the population. In this paper we will focus on the first step, the creation of new temporal categories within one agent. Still, the underlying assumptions is that every act of communication attempts to guide the interpretation process of a hearer to arrive at the topic. Consequently a speaker should provide a hearer with that particular information that discriminates the topic from the other events in the context. Hence, the first task of a speaker who wants to draw a hearer's attention to some topic is to construct a *discriminating* description for it. In order to accomplish this, new conceptual categories might be needed. This allows us to postulate a constructivist learning scheme, the discrimination game [8], by which an agent can invent and adopt new categories. This is in contrast with statistical learning schemes (e.g. clustering algorithms) that attempt to create ontologies independent of the goal of communication. It is also in contrast with attempts to construct standardized and fixed ontologies, be it by hand or otherwise.

The remainder of the paper is organized as follows: first some definitions are

given, including definitions for *discriminating expressions* and for the *discrimination game* (section 2). Next an algorithm is presented for constructing *discriminating expressions* by stating this problem as a general search problem [7] (section 3). Next it is shown how this algorithm can also be used to create new temporal categories (section 4). Finally a discussion and some conclusions follow.

2 The Generalized Discrimination Game, definitions

Context, topic, variables, expressions, bindings and interpretations

In a discrimination game, an agent receives a context C, which is a set of events, and a topic, which is an event in the context. The context is the result of observing the world using some further unspecified observation scheme or event detection algorithm. Every event in the context is described by a predicate. For example a fall event fall_1 is described by the *fall(X)* predicate because *fall*(fall_1) is true. Predicates and variables are written in *italics*, variables also with a capital first letter. Events and other values are written in plain. A binding of a variable X to a value x is written as (X/x). Every conjunction of predicates is called an expression. The substitution of a set of bindings $B = \{(X_1/\text{x}_1), ..., (X_k/\text{x}_k)\}$ in an expression $e(X_m, ..., X_n)$ is written as $[e]_B = e(\text{x}_m, ..., \text{x}_n)$. Finding a set of bindings B for the variables in an expression e to the events in a context C such that $[e]_B$ =true is called interpreting e in C and B is then called an interpretation of e in C.

Discriminating expressions

It could be that several events in the context are described by the same predicate. Assume that the context contains two fall events: fall_1, which happened in the past, and fall_2, which started after the fall_1 event and is still going on (present.) In this case there are two interpretations for the expression *fall(X)*, but only one for the expression $\textit{fall}(X) \wedge \textit{past}(X)$. This last expression is called a *discriminating expression* for event fall_1 with respect to (wrt) the context $C = \{\text{fall}_1, \text{fall}_2\}$ and the variable X.

The expression $\textit{fall}(X) \wedge \textit{before}(X, Y) \wedge \textit{fall}(Y)$ has two interpretations: $B_1 = \{(X/\text{fall}_1), (Y/\text{fall}_1)\}$ and $B_2 = \{(X/\text{fall}_1), (Y/\text{fall}_2)\}$. Still, it is also a discriminating expression for event fall_1 wrt the context C and the variable X because in all interpretations X is bound to fall_1. It is not a discriminating expression for any event in C wrt Y because Y is bound to different events in both interpretations.

Ontology

An ontology is a set of expressions called categories. *fall(X)*, *past(X)* and *before(X,Y)* could be categories. As mentioned, the events in the context are observed by some event detection system. Whether an event is a fall event or another type of event depends on how these types are defined in the event detection system. In this paper, we will assume that the event detection system defines a basic set of predicates like *fall*. As such, it provides an agent with a basic ontology.

It also provides the agent with an event's begin and end times. Categories like

past and *before* call these functions, but are not part of the basic ontology. They have to be created by the agent and are part of the ontology for time.

The discrimination game

A specific attempt to perform a discrimination and the possible subsequent extension of the ontology is called a discrimination game. More formally, given a topic (an event), a context C, with topic $\in C$, and an ontology $\mathcal{O}$, the discrimination game is defined as follows:

> "find an expression $e(X_1, ..., X_n)$, with $n \geq 1$, that is a conjunction of expressions in $\mathcal{O}$, and a variable X_i, with $1 \leq i \leq n$, such that e is a discriminating expression for the topic wrt C and X_i. If this is not possible then $\mathcal{O}$ may be extended with a new expression to make it possible."

3 Searching for discriminating expressions

Constructing a discriminating expression can be formulated as a search problem. As explained in [7] (chapter 3), a search problem is specified by an *initial state*, a *successor function*, a *goal test* and *cost functions* or, equivalently, *score* functions[1].

State definition

A state in the search space is defined as a pair (e, E) where e is an expression and E is a set of expressions. The expression e represents the candidate discriminating expression and the set E contained the remaining categories that were not yet used to build e. Thus, the initial state is $(\mathit{true}, \mathcal{O}')$, *true* being an expression with no variables that is always true and $\mathcal{O}'$ being a subset of $\mathcal{O}$ containing the relevant categories for the current context. For example, if the context does not contain any fall event it makes no sense to include the $fall$ predicate into $\mathcal{O}'$. A category is relevant for a context when an interpretation can be found for it in that context.

Successor function definition

Defining a successor function is not as trivial as might seem. There are two difficulties. First, variables might have to be renamed. Consider the state $(\mathit{fall(X)}, \{\mathit{roll(X)}\})$. The successor function should combine the *roll* expression with the *fall* expression, but clearly the combination $\mathit{fall}(X) \wedge \mathit{roll(X)}$ is different from the combination $\mathit{fall}(X) \wedge \mathit{roll}(Y)$.

The Second difficulty concerns the set E of remaining expressions. Consider the state $(\mathit{true}, \{\mathit{fall(X)}, \mathit{before(X,Y)}\})$. The idea is to combine one of the expressions *before(X,Y)* and *fall(X)* with *true* and remove it from the set of remaining expressions in the resulting successor state. This results in the following two successor states:

$$\{(\mathit{fall(X)}, \{\mathit{before(X,Y)}\}), (\mathit{before(X,Y)}, \{\mathit{fall(X)}\})\} \quad (1)$$

[1] A score function can always be defined as the reciprocal of a cost function or vice versa.

But, continuing in this way, it is impossible to arrive at expression (2) in which the $fall$ predicate is used twice.

Both difficulties can be solved in the following way. Let $s = (e, E)$ be a state for which the successor states have to be constructed. Let the set of variables in e be $\{X_1, .., X_n\}$. Define P to be the set of all possible combinations of length 0 to n of the variables in e: $P = \{\{\emptyset\}, \{X_1\}, ..., \{X_n\}, \{X_1, X_2\}, ..., \{X_1, .., X_n\}\}$.

For every expression $e'(X'_1, .., X'_{n'}) \in E$ and $i \in \{1..n'\}$ define

$$
\begin{aligned}
&m(X'_i, \{\emptyset\}, e') = e',\\
&m(X'_i, \{X_u\}, e') = [e']_{\{(X'_i/X_u)\}},\\
&m(X'_i, \{X_u, X_v\}, e') = [e']_{\{(X'_i/X_u)\}} \wedge [e']_{\{(X'_i/X_v)\}},\\
&\quad ...\\
&m(X'_i, \{X_1..X_n\}, e') = [e']_{\{(X'_i/X_1)\}} \wedge ... \wedge [e']_{\{(X'_i/X_n)\}}
\end{aligned}
$$

Hence, for a given e' and X'_i, $m(X'_i, ., e')$ defines a mapping M over P: $M(X'_i, P, e') \equiv \{m(X'_i, V, e') | V \in P\}$. The set $S_{n'}(e', P)$ of successor expressions resulting from combining e with e' is defined recursively as ($1 \leq k \leq n'$):

1. $S_0(e', P) = \{e'(X'_1, .., X'_{n'})\}$
2. $S_k(e', P) = \bigcup_{e_{k-1} \in S_{k-1}(e', P)} m(X'_k, P, e_{k-1})$

The set $\{(e \wedge e_{n'}, E \setminus \{e'\}) | e_{n'} \in S_{n'}\}$ is a set of successor states of the state (e, E). The set of all successor states is the union of all these sets for every $e' \in E$.

Successor function examples

We will illustrate this with some examples. First, it is obvious that the successor states of the state $(\mathit{true}, \{\mathit{fall(X)}, \mathit{before(X,Y)}\})$ is given by equation (1). Indeed, the set $P = \{\{\emptyset\}\}$ and for *fall(X)* we get: $m(X, P, \mathit{fall}(X)) = \mathit{fall}(X)$. Hence $S_1(\mathit{fall}(X), P) = \{\mathit{fall}(X)\}$, which gives the first element of equation (1). The construction of the second is similar.

For the state $(\mathit{fall(X)}, \{\mathit{before}(X_1, X_2)\})$ the set P is equal to $\{\{\emptyset\}, \{X\}\}$. Hence:

$$
\begin{aligned}
&S_0(\mathit{before}(X_1, X_2), P) = \{\mathit{before}(X_1, X_2)\},\\
&S_1(\mathit{before}(X_1, X_2), P) = \{\mathit{before}(X_1, X_2), \mathit{before}(X, X_2)\},\\
&S_2(\mathit{before}(X_1, X_2), P) =\\
&\qquad \{\mathit{before}(X_1, X_2), \mathit{before}(X_1, X), \mathit{before}(X, X_2), \mathit{before}(X, X), \},
\end{aligned}
$$

giving four successor states. For the state $(\mathit{before}(X, Y), \{\mathit{fall}(X_1)\})$ the set P is equal to $\{\{\emptyset\}, \{X\}, \{Y\}, \{X, Y\}\}$ and there are again four successor expressions resulting from conjugating $\mathit{before}(X, Y)$ with one of the expressions in $S_1(\mathit{fall}(X_1), P) = \{\mathit{fall}(X_1), \mathit{fall}(X), \mathit{fall}(Y), \mathit{fall}(X) \wedge \mathit{fall}(Y)\}$.

Score

To guide and speed up the search process it is customary to associate a score with every state in the search space. A particularly good heuristic here is the minimum number of values left to discriminate from the topic. If this number is d then the score of a state is defined as $1/(1+d)$. Thus, if an expression $e(X,Y)$ has the interpretations $\{(X/\mathrm{v}_1),(Y/\mathrm{v}_2)\}$, $\{(X/\mathrm{v}_2),(Y/\mathrm{v}_1)\}$ and $\{(X/\mathrm{v}_3),(Y/\mathrm{v}_1)\}$, and if v_1 is the topic then $d=1$ (the minimum of 1 (for Y) and 2 (for X)) and the score of the state $S=(e,E)$ is $1/2$. A way to efficiently calculate this score during the search process will be given shortly.

Goal test

The initial state together with the successor function completely determine the search space of possible states. The goal test on such a state (e,E) returns true iff e is a discriminating expression for the topic wrt the context and one of the variables in e, or, equivalently, iff the state has a score 1.

Optimization

Not every expression in the search space necessarily has an interpretation in the context. Those that don't do not need to be considered any further. As mentioned, to construct the initial state, the interpretations of the expressions in $\mathcal{O}'$ are needed. These can be combined along the search path to keep track of the possible interpretations of newly created expressions without having to actually calculate the interpretations in the context. If a newly created candidate expression does not have any interpretations left it is not considered any further.

In addition, the set of interpretations of an expression can be used to efficiently calculate the score of a state without having to calculate the interpretations themselves.

4 Creating temporal categories

Let eT and bT, which stand for *endTime* and *beginTime* respectively, be functions acting on events and returning some time value (a number.) As mentioned, they are part of the interface to event detection system and are not further discussed. Define $\mathcal{R}_t$ as the set of 8 possible temporal relations (predicates) that result from combining these functions with the relations $\leq$ and $\geq$:

$$\mathcal{R}_t = \{r_1(X_1,X_2) = bT(X_1) \leq bT(X_2), .., r_8(X_1,X_2) = eT(X_1) \geq eT(X_2)\}.$$

The relations in $\mathcal{R}_t$ can be used to construct temporal categories. To see this, consider an agent with ontology $\mathcal{O} = \{\mathit{fall}(X)\}$ playing the discrimination game in a context $C = \{\mathrm{fall}_1, \mathrm{fall}_2\}$ with topic fall_1. As before, fall_1 is a past fall event ending before the beginning of the present fall event fall_2. Because $\mathcal{O}$ does not contain any temporal categories yet the agent is unable to find a discriminating expression for the topic. But as can be seen, the expression

$$\mathit{fall}(X) \wedge r_1(X,Y) \wedge \mathit{fall}(Y), \tag{2}$$

is a discriminating expression for fall_1 wrt C and the variable X. Hence, if the agent's ontology is be extended with r_1 he could succeed in the discrimination

game. The expression $r_1(X, Y)$ would then become a temporal category in the agent's ontology.

The algorithm used to search for discriminating expressions can also be used to search for new temporal categories by extending the initial state's set of remaining expressions with the relations in $\mathcal{R}_t$. Thus, when an agent is unable to discriminate the topic using the relevant set of categories $\mathcal{O}'$, he tries again, this time also using the relations in $\mathcal{R}_t$. This is done by setting the initial search state to $(true, \mathcal{O}' \bigcup \mathcal{R}_t)$. If a discriminating expression is found this way, the entire part of the expression that is not yet contained in $\mathcal{O}$ becomes a single new temporal category and is added to $\mathcal{O}$.

5 Discussion and conclusion

In this paper the notions of a *discriminating expression* and of the *generalized discrimination game* were defined. An algorithm for constructing discriminating expressions was presented by defining the problem as a general search problem that can be solved with standard search techniques [7]. A heuristic to speed up the search process was also given.

By playing discrimination games, an artificial agent can detect opportunities to extend his ontology. This happens when the agent fails to construct a discriminating expression for the topic of the game. This is one important difference with other approaches: it is the grounded interaction with the world which triggers the creation of new categories as they are needed. There is no human who decides once and for all on the set of relevant categories.

Finding a new temporal category to extend the ontology can also be done by searching for a discriminating expression but with the ontology extended with basic temporal predicates (the set $\mathcal{R}_t$.) The part of the resulting expression that is not yet included in the agent's ontology is then a new temporal category.

The set of possible temporal ontologies that can be created this way includes the ontologies defined in [1, 3, 6]. There are however 2 important differences with these previous approaches. First, the actual temporal ontology that an agent will evolve is not fixed in advance. It might be equal to Allen's basic temporal categories, but it does not have to be. Again, it is the interaction with the world and with the other agents when communicating temporal information that will decide this. Second, the set $\mathcal{R}_t$ can be easily extended, opening the possibility for temporal ontologies completely different from the ones defined in [1, 3, 6]. For example, by only using the begin and end times of an event to construct the set $\mathcal{R}_t$, one excludes many aspectual categories used in natural language to encode temporal information. But one can easily include additional temporal predicates and relations in $\mathcal{R}_t$ providing other temporal information about an event than its begin and end time.

One might argue that no temporal ontology is created at all but that it is also fixed in advance by defining the set $\mathcal{R}_t$. This is not the case because in our view an ontology cannot be separated from the goal for which it is used or from the environment in which it is used. Because the world continuously changes it is not

possible to define a fixed ontology. This is particularly true when the ontology is used to conceptualize aspects of the world for communication. Since language is a conventional system one cannot define an ontology or conceptualization scheme that is appropriate for all existing languages. For example, not all languages require the same temporal information to be made explicit. In Chinese there is no tense system and temporal information is specified lexically or with aspectual categories. In contrast with this, in Bantu languages it is common to have *several* tense categories for different degrees of remoteness in time, e.g. making a distinction between 'a few days ago' and 'more than a few days ago' [3]. What *is* fixed by defining the set $\mathcal{R}_t$ is the set of *possible* temporal ontologies. Which particular ontology is chosen from this set depends both on the world and on the emergent consensus in the language community. It should be noted that also the search algorithm and in particular the heuristic function used introduce a bias or preference for certain categories over others.

The formalism and algorithm presented in this paper were used by our agents to create a shared temporal ontology. In [2] some first results are reported on this, but it still remains to be investigated whether natural (human like) ontologies are preferred over others and, if so, what the influencing factors are that could explain this.

6 Acknowledgments

This research is partly funded by the European Science Foundation (OMML, CRP 01- JA02: The cultural self-organisation of cognitive grammar.)

References

[1] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] Joachim De Beule. Simulating the syntax and semantics of linguistic constructions about time. *Studies in Language – complementary series*, 2004 (Forthcoming).

[3] B. Comrie. *Tense*. Cambridge University Press, 1985.

[4] Randy J. LaPolla. Language as culture: the conventionalization of constraints on inference. http:////citeseer.ist.psu.edu//215204.html.

[5] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1990.

[6] H. Reichenbach. *Elements of Symbolic Logic*. Macmillan, London, 1947.

[7] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approch*. Prentice Hall, Inc., Upper Saddle River, New Jersey 07458, 1995.

[8] Luc Steels. Perceptually grounded meaning creation. In M. Tokoro, editor, *Proceedings of the International Conference on Multiagent Systems (ICMAS-96)*, pages 338–344, Menlo Park, CA, 1996. AAAI Press.

[9] Luc Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1:169–194, 1998.

[10] James Hendler Tim Berners-Lee and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[11] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 2(10), 1995.