Using Java-Prolog connection to implement automated reasoning *

Nicolae Ţăndăreanu

Faculty of Mathematics and Computer Science,
Department of Computer Science,
University of Craiova, Romania
ntand@oltenia.ro

Abstract. The concept of knowledge base with output was introduced in [12]. In this paper we present a synthesis of some results obtained by implementing a particular case, where the output space is a set of sentences in a natural language. The corresponding method uses a labelled graph and therefore a path driven reasoning is obtained. The inference engine is realized in Prolog and the graphical user interface is realized in Java. In order to realize this application the connection Java-Prolog written by Ugo Chirico is used.
AMS Classification: *Primary: 68T30, Secondary: 68N17, 68N19*
Keywords: *inference engine, labelled graph, labelled stratified graph, Peano algebra, knowledge base with output, Java-Prolog connection*

1 Introduction

Knowledge representation is a central problem and a major area of research in artificial intelligence. The most known methods for knowledge representation can be classified into two main categories. The first one uses the logical representation and the second category uses the graph- based representation of knowledge. There have been a great number of proposals for the semantics of logic representation of knowledge ([1], [4], [5], [17]). In a graph-based representation an entity is given by a pair of nodes and a link between them. The semantics is given by specifying the concrete meaning of this notational convention. In this category can be introduced the following structures: semantic networks, frames, scripts, conceptual graphs. An overview of these methods can be found in ([18]). Many of these methods, both from the first category and the second category, are implemented today ([2], [6], [8], [10], [11]).

The method implemented in this paper has established the starting point in my research work concerning the concept of knowledge base with output ([12]). Several intuitive aspects concerning both the syntactic and semantic computations in a knowledge base with output are described in [14]. We relieve that

* This research was supported by the National Council for Scientific Research of Higher Teaching

in the present paper the output space is a set of sentences in a natural language and therefore the corresponding application can be used in the domain of communicating semantics.

2 Theoretical results

In what follows we shall use the concepts of labelled graph and labelled stratified graph ([12], [13]).

We consider two finite sets S and L_0 such that $S \cap L_0 = \emptyset$. An element of S is called a **node label**; the elements of L_0 are called **arc labels**.

A *binary relation* over S is a subset $\rho \subseteq S \times S$. If $\rho_1 \in 2^{S\times S}$ and $\rho_2 \in 2^{S\times S}$ then we define: $\rho_1 \circ \rho_2 = \{(x,y) \in S \times S \mid \exists z \in S : (x,z) \in \rho_1, (z,y) \in \rho_2\}$. We define the mapping $prod : dom(prod) \longrightarrow 2^{S\times S}$ as follows:

$$dom(prod) = \{(\rho_1,\rho_2) \in 2^{S\times S} \times 2^{S\times S} \mid \rho_1 \circ \rho_2 \neq \emptyset\}$$
$$prod(\rho_1,\rho_2) = \rho_1 \circ \rho_2$$

The mapping *prod* is called the *product operation*. The pair $(2^{S\times S}, \sigma_S)$, where $\sigma_S = prod$ becomes a partial σ- algebra.

Let T_0 be a set of binary relations on S, such that $\emptyset \notin T_0$. Let $f_0 : L_0 \longrightarrow T_0$ be a surjective function. The system $G = (S, L_0, T_0, f_0)$ is called a **labelled graph**.

A labelled graph $G = (S, L_0, T_0, f_0)$ is represented as follows. Every node is represented by a rectangle containing its label. We draw a labelled arc $a \in L_0$ from the node $x \in S$ to the node $y \in S$ if and only if $(x,y) \in f_0(a)$ (Figure 1). Every element of S will designate a node of the graph and only one; thus some bijection between S and the set of the nodes can be established. We say that $a \in L_0$ is a label of the binary relation $f_0(a) \in T_0$. Because f_0 is a surjective function, it follows that every element of T_0 has at least one label.

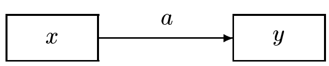

Fig. 1. A labelled arc

Let L_0 be a nonempty set. Let $\mathcal{H} = (H, \sigma_H)$ be the Peano σ-algebra over L_0. For every $L \in Initial(\mathcal{H})$ such that $L \supseteq L_0$ we define the partial σ-algebra $\mathcal{A}_H(L) = (L, \sigma_L)$, where:

- $dom(\sigma_L) = \{(x,y) \in L \times L \mid \sigma_H(x,y) \in L\}$
- $\sigma_L(x,y) = \sigma_H(x,y)$ for every $(x,y) \in dom(\sigma_L)$

The pair $\mathcal{A}_H(L) = (L, \sigma_L)$ is named **the partial σ-algebra associated** to $L \in Initial(\mathcal{H})$.

Let $G = (S, L_0, T_0, f_0)$ be a labelled graph. A **labelled stratified graph over** G is a system $\mathcal{G} = (G, L, T, \sigma_T, f)$ such that:

- $(T, \sigma_T) \in Env(T_0)$
- $L \in Initial(\mathcal{H})$ and $L_0 \subseteq L$, where $\mathcal{H}$ is the Peano σ-algebra over L_0
- $f : \mathcal{A}_H(L) \longrightarrow \mathcal{A}_T$ is a surjective morphism such that $f_0 \prec f$ and if $(f(u), f(v)) \in dom(\sigma_T)$ then $\sigma_H(u, v) \in L$

Because σ is a symbol of arity 2, we have $dom(\sigma_L) \subseteq L \times L$ and $dom(\sigma_T) \subseteq T \times T$. The mapping $f : L \longrightarrow T$ is a morphism, that is, in the diagram

$$\begin{tikzcd}
L \times L \arrow[r, "\sigma_L"] \arrow[d, "f \times f"'] & L \arrow[d, "f"] \\
T \times T \arrow[r, "\sigma_T"'] & T
\end{tikzcd}$$

for every $(u, v) \in dom(\sigma_L)$ the following two properties are satisfied:

$$(f(u), f(v)) \in dom(\sigma_T)$$
$$\sigma_T(f(u), f(v)) = f(\sigma_L(u, v))$$

Let us consider a knowledge piece KP. In order to perform an automated reasoning based on KP, the following steps will be realized:

- the knowledge piece KP is transposed in a labelled graph G and its component S, L_0, T_0 and f_0 are obtained;
- based on [13] some labelled stratified graph (G, L, T, σ_T, f) is chosen;
- some output space O of sentences in a natural language is considered and the semantical computations described in [12] are realized

In order to illustrate the computations we shall consider the following knowledge piece KP: *Peter is a student. John is a schoolboy. Every schoolboy is a presumable student. Peter is John's friend. Peter is George's brother. George is a student.*

Taking the set of nodes $S = \{John, schoolboy, Peter, student, George\}$ and the set of the initial labels $L_0 = \{p, is\_a, pres, is\_brother\}$, the labelled graph $G = (S, L_0, T_0, f_0)$ represented in figure 2 is obtained, where

- $T_0 = \{\rho_1, \rho_2, \rho_3, \rho_4\}$;

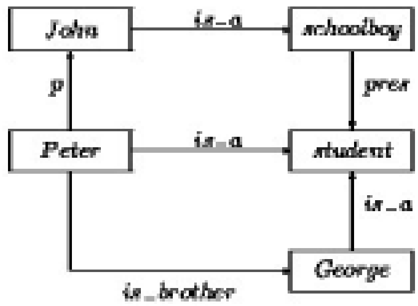

Fig. 2. Labelled graph for KP

- $\rho_1 = \{(Peter, John)\}$, $\rho_2 = \{(John, schoolboy), (Peter, student), (George, student)\}$, $\rho_3 = \{(Peter, George)\}$, $\rho_4 = \{(schoolboy, student)\}$
- $f_0(p) = \rho_1$, $f_0(is\_a) = \rho_2$, $f_0(is\_brother) = \rho_3$, $f_0(pres) = \rho_4$

The closure of T_0 under $prod_S$ is the set $T = T_0 \cup \{\rho_5, \rho_6, \rho_7\}$, where $\rho_5 = \{(Peter, schoolboy)\}$, $\rho_6 = \{(John, student)\}$ and $\rho_7 = \{(Peter, student)\}$. Really, we have:

- $prod_S(\rho_1, \rho_2) = \rho_5$, $prod_S(\rho_2, \rho_4) = \rho_6$
- $prod_S(\rho_3, \rho_2) = \rho_7$, $prod_S(\rho_5, \rho_4) = \rho_7$, $prod_S(\rho_1, \rho_6) = \rho_7$

Taking into consideration the theoretical results exposed in the first part of this section we obtain the following computations:

$f(\sigma(p, is\_a)) = \rho_5$, $f(\sigma(is\_a, pres)) = \rho_6$
$f(\sigma(is\_brother, is\_a)) = \rho_7$, $f(\sigma(p, \sigma(is\_a, pres))) = \rho_7$,
$f(\sigma(\sigma(p, is\_a), pres)) = \rho_7$

Based on these computations we can take for L the set obtained by adding to L_0 the following elements:

$$\{\sigma(p, is\_a), \sigma(is\_a, pres), \sigma(is\_brother, is\_a), \sigma(p, \sigma(is\_a, pres)), \sigma(a, pres)\}$$

On the other hand we observe that

$$f(\sigma(p, \sigma(is\_a, pres))) = f(\sigma(\sigma(p, is\_a), pres)) = \rho_7$$

and the semantics of the label $\sigma(p, \sigma(is\_a, pres))$ is the same as the semantics of $\sigma(\sigma(p, is\_a), pres)$.

In conclusion, to obtain an environment for a labelled stratified graph (see [13]) we can choose the following mapping σ:

$\sigma(p, is\_a) = a$, $\sigma(a, pres) = b$, $\sigma(is\_brother, is\_a) = c$
$\sigma(is\_a, pres) = d$

In the next section we present the manner in which the above results can be used to implement the corresponding reasoning.

3 Implementation

In order to implement a software product based on the results presented in the previous section, the **JIProlog** product written by Ugo Chirico was used (http://www.ugosweb.com/jiprolog). The name JIProlog derives from JavaInternetProlog and this name relieves the fact that a bidirectional connection Java-Prolog is realized. JIProlog is written in Java and the Prolog engine is compatible with other Prolog engines such as SWI Prolog. The user can invoke the Prolog interpreter in any Java applet/application and all the solutions of the Prolog engine can be obtained. A Prolog program can invoke in turn a Java method.

The implementation is based on the following components:

- the inference engine *rsem.pl* written in Prolog
- the component *kp_name.pl*
- the component $Transform.java$
- a graphical user interface $ExtSem.java$

The inference engine *rsem.pl* is able to realize the following:

- find the paths between two nodes of a labelled graph
- for each path computes the associated label
- gives a general structure of the explanation module

The component *kp_name.pl* realizes the following tasks:

- defines the labelled graph associated to a given knowledge piece
- receives the environment, that is the mapping σ
- defines the semantics of the labels corresponding to the knowledge piece taken into consideration

Because a person is a node of the labelled graph and in Prolog such a name begins with a small letter, the component $Transform.java$ receives a person name and transforms the first character in a capital letter. This component, written in Java, is invoked from Prolog to display the conclusion of the reasoning.

The graphical user interface uses the bidirectional connection Java-Prolog and realizes the following tasks:

- ten buttons to perform the following tasks:
  - a button to load the inference engine
  - a button to select some knowledge piece
  - a button to obtain the first conclusion and another button to obtain more conclusions of the reasoning
  - a button to reset the reasoning process
  - a button for help, buttons to display the text of the knowledge piece, the image of the labelled graph and the explanations for the conclusion
- five windows are used to display the conclusion of the reasoning and the explanations, the image of the labelled graph corresponding to the knowledge piece selected by button, the order of the actions, messages and the text of the knowledge piece

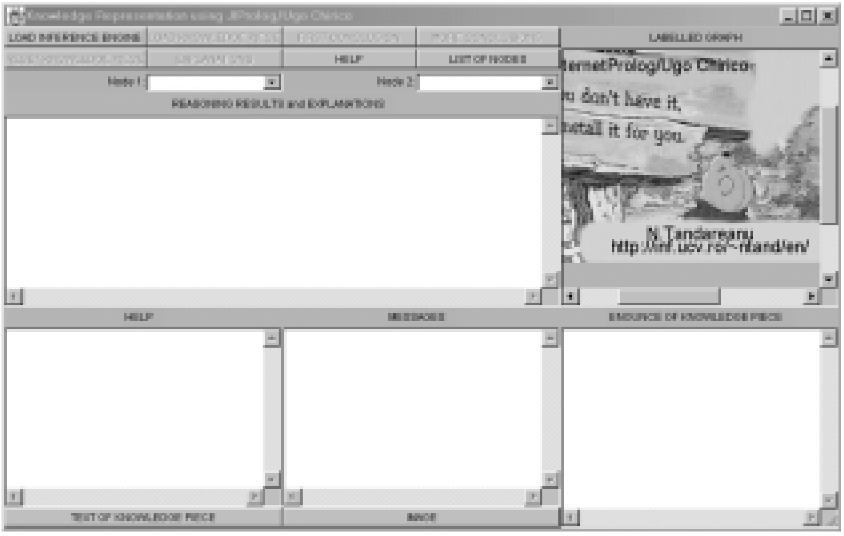

Fig. 3. The initial image of GUI

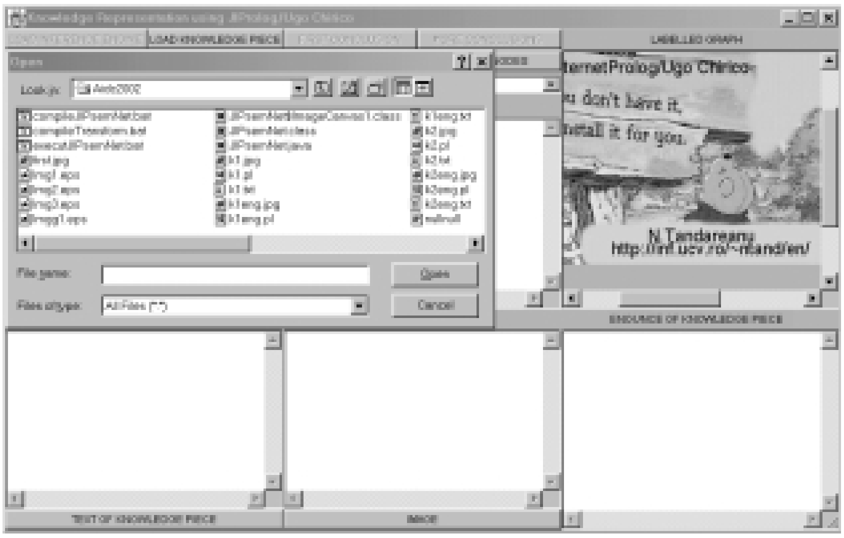

Fig. 4. Selecting a knowledge piece

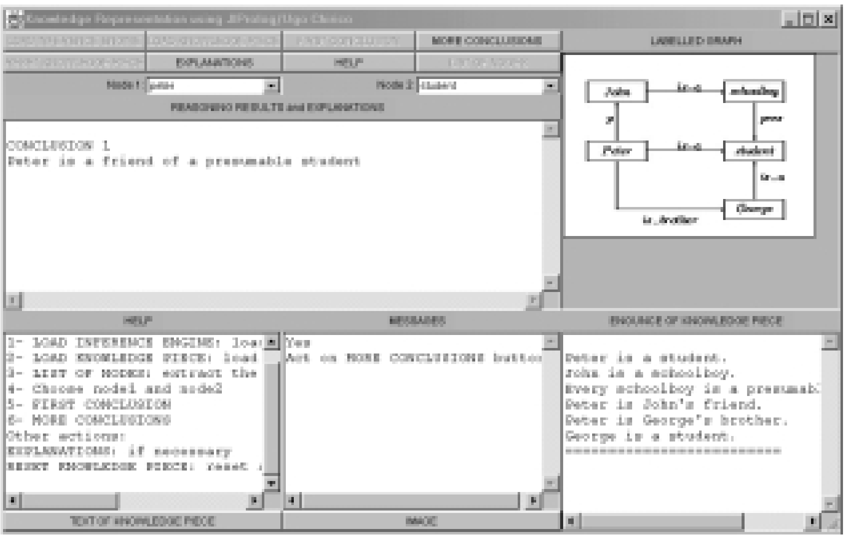

Fig. 5. First conclusion

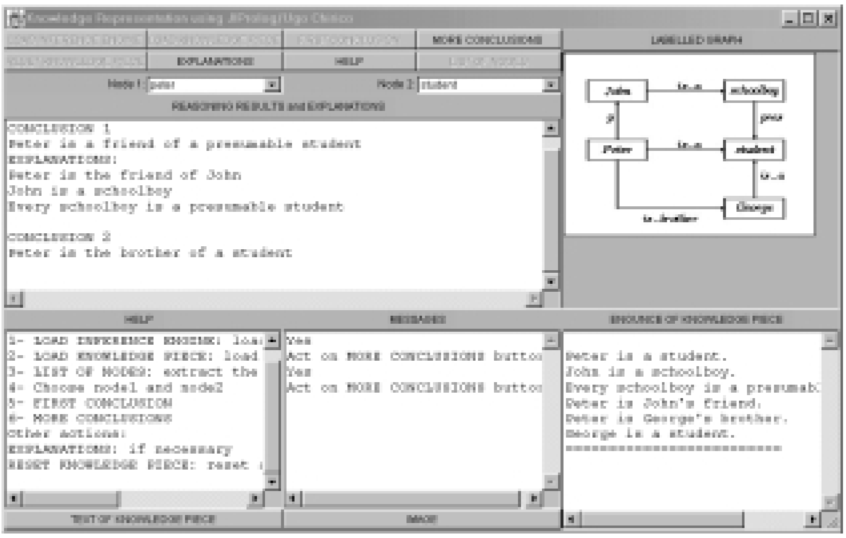

Fig. 6. Conclusions and explanations

If we launch the graphical user interface (GUI), on the computer display the image from figure 3 is displayed.

The next step is the loading of the inference engine by acting the corresponding button. Then a knowledge piece must be selected and the image from figure 4 is shown on the display. The corresponding program in Prolog must be consulted by acting the item *Consult* of a menu. Using the button *LIST OF NODES* we can view all the nodes of the graph and in the next step we select both the first and the second node. By acting the button *FIRST CONCLUSION*, the image from figure 5 is exposed. Then by acting the button *EXPLANATIONS* and then *MORE CONCLUSIONS* the image from figure 6 is lay out.

A complete result of the reasoning corresponding to the first node *peter* and the second node *student* shows as follows:

```
CONCLUSION 1
Peter is the friend of a presumable student
EXPLANATIONS
Peter is the friend of John
John is a schoolboy
Every schoolboy is a presumable student
CONCLUSION 2
Peter is the brother of a student
EXPLANATIONS
Peter is the brother of George
George is a student
CONCLUSION 3
Peter is a student
EXPLANATIONS
Peter is a student
CONCLUSION 4
Finish solutions
```

4 Conclusions and future work

The idea used in this application has constituted the starting point to introduce and formalize the concept of knowledge base with output ([13]). As it is known, such a structure means a labelled stratified graph and an output space endowed with and algebraic operation. On the other hand, a labelled stratified graph uses a partial Peano algebra. Some algebraic morphism with values in the output space gives the semantics of the conclusion of some reasoning. In comparison with a semantic network, the labels are here abstract elements and these elements are "interpreted" in an output space. Various elements may be taken as elements of the output space. For example, the elements may be sentences in a natural language as we proceeded in this paper. The last research was focused on the algebraic properties of the labelled stratified graphs, especially to study so named *the choice problem* ([15], [16]). This problem means the choice of an appropriate labelled stratified graph for a given knowledge piece.

The problem can be avoided by considering the greatest labelled stratified graph for a given context and then by introducing several constraints on the syntactic computations.

The future research in the implementation of these concepts includes the following:

1) consider the labelled stratified graph for a given context and formalize the constraints in the syntactic computations ([15], [16])
consider the case of an output space constituted from pieces of images and study the implications of this choice in image synthesis
consider the case of a non-associative binary operation from the output space

We intend to study the applications of the concepts presented in this paper to problem solving, image synthesis and image analysis.

References

[1] Baral, C. and Gelfond, M. (1994), Logic Programming and Knowledge Representation, *Journal of Logic Programming* **19-20**, 73-148.

[2] Chikayama, T. (1995), "KLIC User's Manual, version 2.001", Institute for New Generation Computer Technology, Japan.

[3] Ugo Chirico: http://www.ugosweb.com/jiprolog

[4] Dahr, M. (1997), "Deductive Databases, Theory and Applications", International Thomson Computer Press.

[5] Gelder, A., Ross, K.A. and Schlipf, J.S. (1991), The Well-Founded Semantics for General Logic Programs, *Journal of ACM* Vol. **38**, No.3, July, 620-650.

[6] Hustadt, U. , Nonnengart, A., Schmidt, R., Timm, J. (1992), "MOTEL User Manual", Max-Planck Institute for Computer Science, MPI-1-92-236.

[7] Thomason, R.H. (1992): NETL and subsequent path-based inheritance theories, Computers Math. Applic., Vol. **23**, No.**2-5**, 179-204.

[8] Rich, C. and Feldman, A.Y. (1992): Seven Layers of Knowledge Representation and Reasoning in Support of Software Development, *IEEE Trans.Software Engineering* **18**, No.6, June, 451-469.

[9] Nado, R., Fikes, R. (1992): Saying more with frames: Slots as classes, Computers Math. Applic., Vol. **23**, No.**6-9**, 719-731.

[10] Shapiro, S.C. and Rapaport, J.W. (1992), The SNEPS Family, *Computers Math. Applic.* **23**, No.2-5, 243-275.

[11] Shell, P. and Carbonell, J. (1991) "PARMENIDES: A Class-Based Frame System, version 1.5", Carnegie Mellon University.

[12] Ţăndăreanu N. (2000), Knowledge Bases with Output, Knowledge and Information Systems, 2, 438-460.

[13] Ţăndăreanu N. (2000), Proving the existence of labelled stratified graphs, Annals of the University of Craiova, Mathematics and Computer Science Series, XXVII, 81-92.

[14] Ţăndăreanu N. (2001), Intuitive aspects of the semantic computations in a KBO, Research Notes in Artificial Intelligence and Digital Communications, Vol.101, RCAI 1st Conference on Artificial Intelligence and Digital Communications, Craiova, p.1-8.

[15] Ţăndăreanu N. Ordering environments over a labelled graph (to appear).

[16] Ţăndăreanu N. The greatest LSG over a labelled graph (in preparation)
[17] Wagner, G. (1994): Vivid Logic, Knowledge-Based Reasoning with Two Kinds of Negation Springer Verlag, Lecture Notes in Artificial Intelligence **764**.
[18] Way, E.C. (1991), "Knowledge Representation and Metaphor", Kluwer Academic Publisher, Studies in cognitive systems, Vol. **7**.