Sorin Marian DINCĂ

KNOWLEDGE BASES MODELED BY SEMANTIC SCHEMAS AND THEIR QUERIES

Sorin Marian DINCĂ

KNOWLEDGE BASES MODELED BY SEMANTIC SCHEMAS AND THEIR QUERIES

Editura UNIVERSITARIA
Craiova, 2016

Referenți:

Prof.univ.dr. Tudor BĂLĂNESCU

Lect.univ.dr. Claudiu Ionuț POPÎRLAN

Descrierea CIP a Bibliotecii Naţionale a României
DINCĂ, SORIN
Knowledge bases modeled by semantic schemas and their queries / Sorin Marian Dincă. - Craiova : Universitaria, 2016
Conţine bibliografie
ISBN 978-606-14-1004-0

004

List of Figures

cooperating systems of structures based on semantic schemas were introduced in literature ([50], [51], [52]). Several applications based on these structures were described and implemented today in the domain of distributed knowledge and dialog systems ([58], [62], [73], [74], [84], [85], [86]). Hierarchical structures of semantic schemas were applied to image generation ([49], [55], [57]).
I can say that the results presented in this book belong to a common domain of computer science and mathematics because the methods of the universal algebras are fully imputed to formalize the computations accomplished by semantic schemas. Significant results were obtain in collaboration with *Prof. Dr. Nicolae Ţăndăreanu* [54].
The concept of semantic schema was introduced as a method of knowledge representation. Two kinds of computations were considered in such a structure: formal computations and semantic computations. The formal computations are described by means of a Peano algebra obtained by means of the components of the semantic schema. Some of these elements are interpreted by means of an interpretation. This means that the abstract elements of such an entity become objects of a real world (sentences, geometric images etc).
The main objectives of this study are the following:

1. To redefine the formal computations performed by the inference engine of a semantic schema starting from a concept introduced by Prof. Dr. Nicolae Ţăndăreanu. The old formal computation, introduced in [48] by Prof. Dr. Nicolae Ţăndăreanu, was unproductive one because a non-deterministic computation was obtained. More precisely, I consider the Peano σ-algebra generated by the elementary arcs of a semantic schema $\mathcal{S}$. Some elements of this algebra are the useful elements of the inference process of $\mathcal{S}$. I propose to identify in a deterministic manner all these elements.

2. I reconsider the concept of structured path in semantic schemas introduced in [50]. Based on the results obtained by previous stage of the research, I redefine the formal computations by structured path in order to optimize knowledge base representation.

3. To apply the results of the research to model the distributed knowledge representation and to use these results in the domain of the dialog systems.

The results presented in this book contribute to redefine the computations in semantic schemas and develop, as a consequence, the computations in cooperating systems based on these structures. I discuss also some applications (*Java application for distributive computations using semantic schemas* and *Developing a system of knowledge bases* *using semantic schemas in Java and CLIPS*) and the corresponding implementations of the structures presented in the chapters of the book.

Chapter 2

Structured paths and relations in semantic schemas

2.1 Introductory elements

In this chapter I give a description of the structured path starting from the research made by *Prof. Dr. Nicolae Ţăndăreanu* in his papers [45], [46], [47].
The direct ordered graphs are defined and some examples are given for ordered trees and θ-schema. Starting from this notions, I describe the structured paths and I also give examples used to exemplify the notions introduces in this chapter. The examples used in this chapter permit me to analyze the construction of every elements that are used in a structured path.
For the examples presented in this chapter visual representations are given. The figures are intuitive and are used for a better understanding of the notions introduced in the chapter.
A *directed ordered graph* ([1]) is a pair $G = (A, D)$, where

- A is a finite set of elements called *nodes*
- D is a finite set of elements of the form $[(i, i_1), \ldots, (i, i_n)]$, where $n \geq 1$ and $i, i_1, \ldots, i_n \in A$
- D satisfies the following condition: if $[(i, i_1), \ldots, (i, i_n)] \in D$ and $[(j, j_1), \ldots, (j, j_s)] \in D$ then $i \neq j$

I observe that for an element $[(i, i_1), \ldots, (i, i_n)] \in D$ we may have $i_j = i_k$ for some $j \neq k$. On the other hand, an element of D is a list and the *order* of its elements are

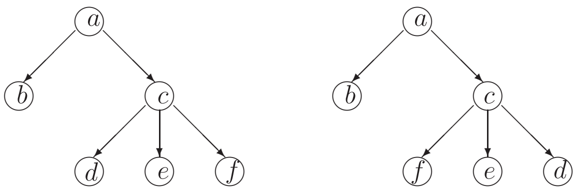

Figure 2.1: Two distinct ordered trees

taken into consideration. An element of a list is a *directed arc* and simply is named *arc*. This can explain why the concept is named *directed ordered graph*.

I can represent a directed ordered graph ([63], [64], [65]) as follows. I represent, as usual, a node of the graph by a point. If $[(i, i_1), \ldots, (i, i_n)] \in D$ then I draw *an arc* from node i to node i_j for every $j \in \{1, \ldots n\}$. The elements $i_1, \ldots, i_n$ are called *the direct descendants* of i. I shall consider that all direct descendants of i are ordered linearly and the order is given by the place of i_j in the element $[(i, i_1), \ldots, (i, i_n)]$.

If $G = (A, D)$ is a directed ordered graph [63] then I can associate to G a *directed graph* $G' = (A, D')$, where

$$D' = \{(i, j) \mid \exists [(i, i_1), \ldots, (i, i_n)] \in D, \exists r \in \{1, \ldots, n\} : j = i_r\}$$

An *ordered tree* [65] is a directed ordered graph $G = (A, D)$ such that G' is a tree and the following property is satisfied:

$$[(i, i_1), \ldots, (i, i_n)] \in D, j, r \in \{1, \ldots, n\}, j \neq r \Rightarrow i_j \neq i_r \tag{2.1}$$

Two distinct ordered trees are represented in Figure 2.1. In the left part of this figure I have $[(c, d), (c, e), (c, f)] \in D$, whereas in the right part I have $[(c, f), (c, e), (c, d)] \in D$. They have the same set of nodes but the order is another.

A *path* in a directed ordered graph [64] is a sequence $d = (n_0, n_1, \ldots, n_k)$ of nodes such that for every $i \in \{0, \ldots, k-1\}$ I have an arc from n_i to n_{i+1}. The number k is the *length* of d. I denote by $Path(G)$ the set of all paths in G.

Consider a θ-schema $S = (X, A_0, A, R)$ and a symbol h of arity 3. Let us denote

$$M = \{h(x, a, y) \mid (x, a, y) \in R_0\}$$