

Modeling knowledge base and derivation without predefined structure by Graph-based Clausal Form Logic

Martin Žáček, Alena Lukasová, Rostislav Miarka

University of Ostrava, Faculty of Science, Department of Informatics and Computers,
30. dubna 22, 701 03 Ostrava, Czech Republic

Abstract

This paper follows the work of T. Richards specialization Clausal Form Logic formal system of the first order logic. The paper presents also the way of using graph-based clausal form statements in the frame of semantic (associative) networks. The goal of our research is to follow the direction towards graph-based clausal form knowledge representation shaped by Richards and build up a graph-based formal system. The new formal system Graph-based Clausal Form Logic has its own graph-based language with the expressivity similar to that one of CFL. As an example solution, a logical puzzle is presented.

Keywords: Knowledge representation, modeling, knowledge base, formal system, Clausal Form Logic, graph.

1. Introduction

T. Richards shows the way of using graph-based clausal form statements in the frame of semantic (associative) networks. Richards [1] uses the graph-based representation especially for the illustration of meaning of clausal form expressions of CFL.

The goal of our approach is to follow the direction towards graph-based clausal form knowledge representation shaped by Richards, and build up a graph-based

formal system that does not only graphically illustrate knowledge bases, but also allows users to obtain consequents of a knowledge base in a graph-based way.

The new formal system GCFL (Graph-based Clausal Form Logic) has its own graph-based language with the expressivity similar to that one of the CFL; moreover it uses the inference methods of associative networks [2] and proposes a graph-based modification of the resolution inference method of reasoning corresponding to that one of the CFL.

The goal of this paper is mainly to establish and verification of the theory in practice and the most difficult way, which is a logical puzzle (the Einstein logic puzzle – with predefined structure), but we don't have predefined structure of GCFL and derivation is more complex.. The authors introduce a version of Graph-based Clausal Form Logic [5] based on associative networks. GCFL is based on rules of resolution and we introduced the classical derivation by this resolution rule in an example, but also his version graph.

The idea of the GCFL graph-based approach is also useful in the frame of the RDF model especially within its graph version. To create RDF(S) knowledge base using the GCFL language completed by corresponding URIs is not difficult comparatively with an approach of OWL language representation. Finally the GCFL serves an own easy-to-understandable and usable inference mechanism.

2. Richard's Clausal Form Logic (CFL) and its Graph-Based modification (GCFL)

As one of the main tools of formal reasoning, the CFL introduced by T. Richards [1] uses the conditional „if – then“ statements.

A simple example illustrates the “if – then” statement structure by a “Holmes rule”:

*“If one person x hates another person y,
then x knows y.”*

or

If hate(x,y) then know(x,y)

or

If <antecedent> then <consequent>

Generally a conditional statement (clause) proposed by T. Richards says that some (composed) consequent statement follows from another (composed) antecedent statement.

Richards also proposed an alternative representation of the clausal form atoms (vectors) in a graph-based language well known in associative (semantic) networks.

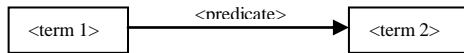


Fig. 1 Vectors

Our idea of graph based representation of clauses connects to that one of Richards' CFL and proposes a complete graph-based inference system GCFL. Knowledge base of the system consists of clauses represented by graphs.

To distinguish statements in the antecedent part of a graph-based clause from statements in its consequent part we introduce a convention

- to draw the arcs of antecedent vectors by *dashed lines* and
- to draw the arcs of consequent vectors by *solid lines*.

All the vectors (with solid or dashed lines) represent atomic statements and have generally the structure

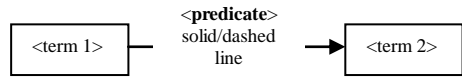


Fig. 2 The structure of the vectors

In the Richards' clause language a general formula of CFL is of a form

$$\begin{aligned} &\langle \text{antecedent} \rangle \rightarrow \langle \text{consequent} \rangle \\ &\text{or} \\ &P_1 \& \dots \& P_m \rightarrow Q_1 \vee \dots \vee Q_n \quad (1) \\ &\text{or} \\ &P_1, \dots, P_m \rightarrow Q_1, \dots, Q_n \end{aligned}$$

where “ \rightarrow ” is a meta-symbol, the antecedent is a conjunction of some set of positive first order logic atoms (vectors) $\{P_1, \dots, P_m\}$ and the consequent is a disjunction of another set of positive first order logic atoms (vectors) $\{Q_1, \dots, Q_n\}$. The convection of two mutually different arc lines in the GCFL does not need to separate antecedent and consequent by any meta-symbol like “ \rightarrow ”, all the dashed antecedent vectors are mutually connected by $\&$ and all the solid consequent vectors are mutually connected by \vee .

In our Holmes example presented above we draw a vector with dashed-line arc as the antecedent part of graph representing a clause

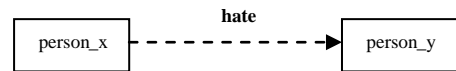


Fig. 3 Antecedent

and as the consequent part of the graph a vector with solid-line arc

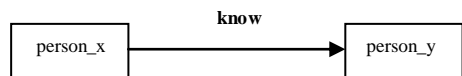


Fig. 4 Consequent

So the Holmes example above has in the graph language GCFL a form a clause network

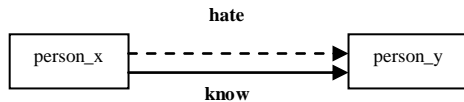


Fig. 5 The clause network

An antecedent or a consequent of the conditional clause in the GCFL can be also an empty set of atoms like in the same way as in the CFL.

In the case of an empty antecedent in the CFL for example the atom

→ **know**(person_x, person_y)

has a meaning of a *positive fact* “A person_x knows a person_y.”. To express is using the syntax of the graph language of GCFL a vector like that one at the Fig.4 suffices.

In the case of an empty consequent in the CFL the formal representation is of the form

know(person_x, person_y) →

that represents a negative fact “It is not true that a person_x knows a person_y.”. To represent is using GCFL a vector with dashed line arc suffices (Fig.6).

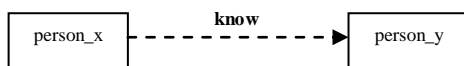


Fig. 6 The structure of the vectors

The structure of the clause allows only constructions of clauses with connections & in the antecedent and connection ∨ in the consequent. If necessary GCFL as well as CFL solves the problem of disjunction in the antecedent (conjunction in the consequent) by a following decomposition of the clause into m (n) separate clauses:

$$\begin{aligned} P_1 \rightarrow Q_1 \vee \dots \vee Q_n & \quad P_1 \& \dots \& P_m \rightarrow Q_1 \\ P_m \rightarrow Q_1 \vee \dots \vee Q_n & \quad P_1 \& \dots \& P_m \rightarrow Q_n \end{aligned} \quad (2)$$

3. Resolution reasoning on GCFL knowledge base

GCFL works by the help of two rules - the substitution rule and the cut rule, both are well known in the first order logic. Together both rules form the resolution rule as well as in the CFL. In the frame of associative networks [2, 3] the resolution rule has been slightly modified and it is known as the transfer rule.

The transfer rule

Conditional clause is a tool of transferring its consequent into another clause unfixable with its antecedent. It is possible (for example) to use the isa rule (Fig.6) for transferring its consequent (after proper substitution into another network).

The substitution rule:

A new clause can be obtained from a clause with variables by a uniform substitution of a term for some of the variables.

The cut rule:

If there are two clauses sharing the same atom in the knowledge base for reasoning, one in the antecedent of the first clause and one in the consequent of the second clause, then we can obtain a new clause by cutting out the same atoms at both sides and create a new clause with antecedent (consequent) that contains all the atoms of the original clauses antecedents (consequents).

4. Logical puzzle

We have this example:

The three artists met in a cafe: Mr. White is a sculptor, Mr. Black is a pianist and Mr. Rusty is a painter.

“It is noteworthy that one of us is white-haired (white), one black-haired (black) and one rusty-haired (rusty), but none of us has a color of hair, which shows his own name” said man with black hair.

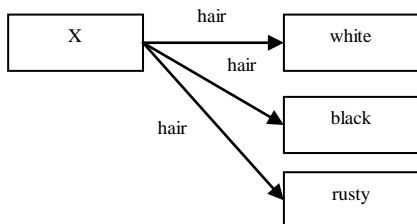
“You are right.” Mr. Whiteanswered.

What color hair has a painter?

Creating a knowledge base

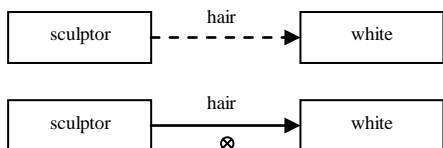
Demonstration 1

→ **hair**(X, white), **hair**(X, black),
hair(X, rusty)



Demonstration 2

hair(sculptor, white) →



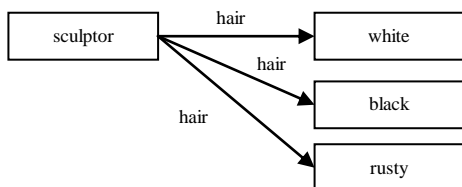
We can rewrite any network with antecedent by negation to a network with consequent. In our case it is unwanted.

Deduction

Demonstration 3

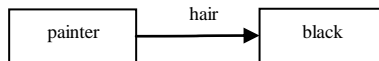
substitution {sculptor/X} into 1.

→ **hair**(sculptor, white),
hair(sculptor, black),
hair(sculptor, rusty)



The result of the derivation by rules

→ **hair**(painter, black)



5. Acknowledgment

The research described here has been financially supported by University of Ostrava grant SGS23/PřF/2013 and 1184-Volná Podpora VaV. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

6. References

- [1] Richards, T.: Clausal Form Logic. An Introduction to the Logic of Computer Reasoning. Addison-Wesley, 1989.
- [2] Lukasová A.: Knowledge representation in associative networks (in Czech) Proceedings of Znalosti 2001. Praha, 2001.
- [3] Lukasová A., Telnarová Z., Habiballa, H., Vajgl, M.: Formal knowledge representation (in Czech). Universum, 2010. 345 p.
- [4] Lukasová, A., Vajgl, M., Žáček, M.: Reasoning in RDF graph formal system with quantifiers. Proceedings of the International Multiconference on Computer Science and Information Technology. 2010. pp. 67-72.
- [5] Lukasová, A., Žáček, M., Vajgl, M. „Reasoning in Graph-based Clausal Form Logic“. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3 (2012) pp. 37-43. ISSN (Online) 1694-0814.