FUNCTIONAL REPRESENTATION OF KAUPPI'S CONCEPT OPERATIONS AND CONCEPT ASSOCIATIONS

Marko Junkkari and Marko Niinimäki

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TAMPERE

REPORT A-1998-12

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS A
A-1998-12, NOVEMBER 1998

FUNCTIONAL REPRESENTATION OF KAUPPI'S CONCEPT OPERATIONS AND CONCEPT ASSOCIATIONS

Marko Junkkari and Marko Niinimäki

University of Tampere
Department of Computer Science
P.O. Box 607
FIN-33101 Tampere, Finland

ISBN 951-44-4473-6
ISSN 0783-6910

Functional representation of Kauppi's concept operations and concept associations

Marko Junkkari

Marko Niinimäki

University of Tampere

Department of Computer Science

November 25th, 1998

Abstract. In conceptual modelling we need exact and systematic methods and equipment to find a relevant and an accurate model of the Universe of Discourse. To manage concepts we need relations and operations that associate them. In this paper, we present the covering set of associate relations and concept operations. By using these we can search concepts that are in some way in association with one or several concepts in a concept system. The main associations with two or several concepts are compatibility and comparability.

The operations and relations that we present here are based on Kauppi's concept theory. There is only one two-placed basic relation in Kauppi's theory, the relation of intensional containment. We use set theory as the meta-language for concept theory, since set theory is an established and well-known manner for formal representation in computer science. Using set theory, we build functions, some of which utilise paths in sets of concepts. By using this approach we also create a system for an explicit representation of Kauppi's theory.

1. Introduction

Concepts are usually considered as results of generalisation from sets of individuals or from other concepts [see Smith and Smith, 1977]. In this way it is natural to take the subsumption relation [Woods, 1991] to be the main relation for presenting containment among concepts. According to this approach, the containment among concepts are based on inclusion of individuals' sets, and the conception of concepts is extensional. This conception is limited, because if there exists no extension of a concept, there is no concept at all. Contrary to this conception, we understand concepts to be discrete knowledge units that subsist relatively independently [see also: Sowa, 1984].

Raili Kauppi was a philosopher, who researched Leibniz' Logic. Kauppi's principal work was the intensional concept theory [Kauppi, 1967]. Unlike many others, Kauppi does not define the intension of a concept as a set of attributes. Instead, she takes an undefined relation, *intensional containment*, on which the intension of concepts is based. It is said that intensional containment covers many relationships between concepts, for example ”is-a”, ”has” and ”contains” [Kangassalo, 1996].

Our contribution here is to describe a large part of Kauppi's theory and form an abstract implementation of it. For its representation we use standard set theory and our formalism is based on Niemi and Järvelin's [1992] paper. Järvelin and Niemi [1993] have presented that this kind of an approach is suitable for management of relationships, which are very close to the well-known is-a-relation.

There are only two ontological primitives in Kauppi's system: concept and non-logical intensional containment relation that is a binary relation among concepts. If concept a contains concept b intensionally, we can also say that b is a characteristic of concept a.

In the following version of concept theory there are three axioms to start our system with. We consider that the relation of intensional containment is reflexive, transitive and antisymmetrical. In our system, the reflexivity and transitivity of the relation are presented in an implicit way. In other words we consider only immediate containment [see Junkkari and Niinimäki, 1998]. For example, if concept a contains intensionally concept b and again b contains intensionally c, we present explicitly only these relations, but not the one that concept a contains intensionally concept c (transitively).

The other axioms of Kauppi's concept theory determine more the structure of the concept system. We do not assume all those axioms, because we want to get a more opulent structure in the set of concepts. Some of these axioms, as well as functions to check the legality of a concept system are presented earlier in [Junkkari and Niinimäki, 1998]. Here and there, we consider only finite concept systems. Here, we also present some expansions to the concept theory, namely that the result of the functions can be a set of traced concepts instead of a unique concept. According to our approach, we get sensible results of the operations, although the structure of the concept system, at hand, does not comply with such a strict axiomatization as in the original theory.

In this paper, we understand that a concept system is a set of concept pairs <a,b> such that concept a immediately contains concept b intensionally. This set is also our basic relation I-rel. By using this relation, we can form directed paths from one concept to another. There are two notable sets of concepts in a concept system, *specie* and *genuses*. If a concept inheres in the set specie, this concept can only be a starting point in any directed path of the concept system. Respectively, if a concept inheres in the set genuses it can be only the end point in any directed path.

Kauppi often writes about the greatest (Ger.: grösste) or the least (Ger.: kleinste) concept in a collection of concepts. The greatest concept means the concept, which contains intensionally all the other concepts in the collection that is created using some relevant principles. Because our approach is set-oriented, we call a maximal concept the concept that is not contained in any other concept as itself in a given set of concepts. Contrary to Kauppi's system, there can be several maximal concepts in a concept set. Analogously, we say that a concept is minimal (least), if there is not any concept that contains intensionally this concept in a given set. The sets of minimal concepts and maximal concepts have a significant role in our representation.

There are eight association relations between two concepts. We say that the concepts a and b are *compatible*, if and only if there is a concept c that contains both a and b intensionally. If c is contained in both the concepts a and b, we say that concepts are *comparable*. The opposites of these are incompatible and incomparable. By combining these four relations we get four more association relations that are *homogen-compatibility*, *heterogen-compatibility*, *opposition* and *isolation*. We also define the functions that return the set of concepts, which are in a certain association with concept a.

When we consider the relationships between one or two concepts and their associations to whole concept system, we talk about concept operations. These are *intensional product*, *intensional sum*, *intensional negation*, *intensional reciprocal*, *intensional quotient* and *intensional difference*.

Intensional product and intensional sum are three-placed relations in the concept system. The third member of the relation is called the result of the operation. Intensional product applies to common characteristics and intensional sum applies to concepts, which contain both the operand concepts intensionally.

The principle of concept operations is to restrict the set of concept using some relevant principles, as containment relation and association relations, and consider maximal or minimal concepts under the restrictions. In our consideration we first form the ”restricted” set of concepts and then we find the maximal or minimal concepts in this set. In a concept system, if there is a concept to correspond the result of the concept operation in the sense of Kauppi, our functions return the set where there is exactly one concept.

By using functions and set theory as tools of representation, we present in an explicit and well-known manner the formal concept theory, which we see to have potential to manage knowledge and to describe the structure of knowledge. In this paper, we generally consider concept theory in a very formal level and without any inherent ontological or semantic attachments.

In section 2, we consider the position of this paper in research of computer science. In section 3, we present our notations and in section 4 we introduce the basics of our set-oriented approach. Simple intensional relations are introduced in section 5 and the functions of association relations are presented in section 6. Concept operations are introduced in sections 7-9. Kauppi’s original definitions of the same operations are presented in the appendix.

2. Motivation

The representation of the structure of information and knowledge has become important in many areas in computer science. For the purposes of the representation, we need to find relevant information and make a structural model of it [Kangassalo, 1993]. Consequently, we need conceptual and logical methods of modelling. A concept theory forms a firm basis for these methods and intensionally oriented conceptual modelling.

The effort for finding relevant information and representing its structure is connected with AI (artificial intelligence), too. For example in DL (description logics) [see e.g. Borgida *et al,* 1989; Borgida, 1995] the knowledge structure is generated by classification of individuals and forming is-a -relationships and roles between class concepts. In DL, the part-whole relations have been a recent object of interest and dispute [Lambrix, 1996; Artale, Franconi and Guarino, 1996; Artale, Franconi, Guarino and Pazzi, 1996]. On the basis of the concept theory, we propose a system that can embed other partial order relation like component relation that is one kind of a part-whole relation. According to Kangassalo [1996] intensional containment can present the component relation, too.

In DL and in semantic networks [see e.g. Brachman and Schmolze, 1985], one primary portion of research has been to find a framework for representation of knowledge. A recent contribution to knowledge representation has been formal ontology and different ontological systems. An ontology makes the basis for explicit presentation of knowledge. In general an ontology is a meta-level description of knowledge representation [Guarino, 1997].

The relationships between Kauppi's [1967] concept theory and ontological issues can be described as follows. On one hand, we consider that concepts themselves are discrete units of the knowledge we try to describe [Sowa, 1984]. On the other hand, Kauppi's theory itself is an ontological system with two ontological primitives: concepts and the relation of intensional containment. This does not mean that for example individuals, properties, attributes, relations and so on are left outside the theory. On the contrary, on the ontological point of view this can be understood so that these are included in the theory implicitly. We see this so that an object that is presented by a concept in Kauppi's theory can appear as a class concept, an attribute, a relation and so on in some other ontological system.

Kauppi designed a concept calculus with a substantial set of operations and association relations in a collection of concepts. We can compare Kauppi's calculus with DL, if we equate the relation of intensional containment with the is-a relation. Then, for example, the operations of intensional sum and intensional product are counterparts for the operations AND and OR in DL. Using the association relations, one can study various relationships (like comparability and compatibility) between concepts: if two concepts are compatible there exists a concept that is a specialisation of both the concepts. To our knowledge, an approach involving concept associations has not been applied in DL.

Kauppi's concept theory can be seen as a minimal covering ontology, wherewith it is possible to present the structure of knowledge. It is clear that a system that follows an ontology of this kind is not practical as such. It requires further study to find the parts of the theory that are applicable in knowledge representation. Our contribution in this paper is that we have presented in an explicit and detailed way the abstract implementation of a covering set of concept operations and association relations of concept theory.

3. Basic notations

Our formalism is based on the formalism that Niemi and Järvelin [1992] use.

1) A set is a collection of elements.
2) The power set of the set S includes all subsets of S as elements and it is denoted by P(S). For example if S = {a, b, c}, then P(S) = {{}, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}}.
3) A finite n-tuple is an organized n-place queue of elements and it is denoted between angle brackets, for example <a,b,c>. A 2-place tuple is also called an ordered pair. The empty tuple is denoted by <>.
4) Binary relation is a set of ordered pairs. If a relation R is subset of Cartesian product $S \times S$, we say that relation is on the set S. If an ordered pair <a,b> accomplish the relation R, this denoted by <a,b> $\in$ R. In an ordered pair, the first member is called predecessor and the second member is called successor.
5) The tuple set of a set S is denoted by T(S). A tuple is an element of T(S) if all elements of the tuple are different and they are members in the set S. Tuple set T(S) is composed of all permutations of the power set P(S) and present these as tuples. For example, if S = {a , b, c} then the T(S) is {<>, <a>, <b>, <c>, <a,b>, <b,a>, <a,c>, <c,a>, <b,c>, <c,b>, <a,b,c>, <a,c,b>, <b,a,c>, <b,c,a>, <c,a,b>, <c,b,a>}
6) The signature of a function *f* is denoted by *f*: S1 $\rightarrow$ S2, where S1 defines a set of values to which the function can be applied and S2 defines the set of values, which the result of the function inheres in.

4. Concept system and the primary functions

C-set is the finite set of concepts related to a specific concept system. As an example, we consider the set C-set1 = {G, gadget, radio, clock, clock radio}, where G is a so-called general concept that is contained intensionally in every concept. Except in C-set1, we do not assume the existence of a single general concept in a set of concepts.

In each concept system, there is an intensional relation I-rel on the set C-set. This relation is a binary relation that corresponds to immediate containment among two concepts [see: Junkkari and Niinimäki 1998]. I-rel is a subset of the Cartesian product C-set $\times$ C-set. The interpretation of I-rel is: if $<a,b> \in$ I-rel, concept a immediately contains the concept b intensionally.

As an example, the relation I-rel1 on C-set1 is the following binary relation: I-rel1 = {<clock radio, radio>, <clock radio, clock>, <radio, gadget>, <clock, gadget>, <gadget, G>}. This is presented in the figure 1, whose illustration method is based on Concept D [Kangassalo, 1993], where an upper level concept contains lower level concepts intensionally.

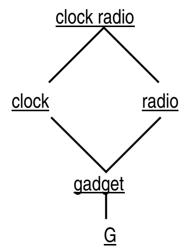

Figure 1. I-rel1

In general, the I-rel alone does not provide sufficient information about the concept system. Suppose, for example, that there are only two concepts, c1 and c2, and neither of them contains the other. In this case, I-rel would be an empty set. Yet, this kind of concept system is different from a concept system that has no concepts at all. Therefore, we define the concept system to consist of both a C-set and an I-rel. We assume that a C-set is associated with each I-rel. We compute transitive and reflexive relationships using I-rel, if we need them. This can be presented for example by closures [Junkkari and Niinimäki, 1998], but here we use a more explicit method based on paths. A simple directed path (or a path for short) is presented as a tuple of concepts $<a_1,a_2,..,a_n>$, where for every $i \in \{1, \ldots, n\}$ holds that $<a_i,a_{i+1}> \in$ I-rel.

The function *path_set* generates all paths from concept a to concept b in the given concept system I-rel. The function takes three arguments: two concepts, a and b, and a two place relation I-rel. Based on the ordered pairs which are included in the I-rel, we construct all the paths from a to b (see operation 15[1] in [Niemi and Järvelin, 1992]).

path_set: C-set × C-set × P(C-set × C-set) → P(T(C-set))

path_set(a,b,I-rel) =

$$\begin{cases} \{< a_1, .., a_n > \mid < a_1, .., a_n > \in T(\text{C-set}) : a_1 = a \wedge a_n = b \wedge \\ \quad \forall i \in \{1, .., n-1\} : < a_i, a_{i+1} > \in \text{I-rel}\}, \text{if } a \neq b \\ \{< a >\}, \text{if } a = b \end{cases}$$

Now in the example *path_set*(clock radio,G,I-rel1) is the set {<clock radio, clock, gadget, G>, <clock radio, radio, gadget, G>}.

We define two auxiliary functions, *maximal_set* and *minimal_set*, for further purposes for finding intensionally maximal and intensionally minimal concepts from a given subset of all the concepts in a concept system. Given a set of concepts C-set' (⊆ C-set), an intensionally maximal concept is such that in C-set' there is no concept that intensionally contains it. Respectively, an intensionally minimal concept is such that in C-set' there is no concept that is intensionally contained in it. The defined functions yield the set of maximal concepts and minimal concepts from given concept set C-set':

maximal_set: P(C-set) × P(C-set × C-set) → P(C-set)

*maximal_*set(C-set',I-rel) = {x ∈ C-set'| ¬∃y ∈ C-set': <y,x> ∈ I-rel}

minimal_set: P(C-set) × P(C-set × C-set) → P(C-set)

minimal_set(C-set',I-rel) = {x ∈ C-set'| ¬∃y ∈ C-set': <x,y> ∈ I-rel}

In figure 1, *maximal_set*({clock, radio, gadget},I-rel1) returns the set {clock, radio}, whereas *minimal_set*({clock, radio, gadget}, I-rel1) returns {gadget}.

---

[1]Niemi and Järvelin have taken the relation base as an argument, too. We have altered the operation so that if a and b are identical, the function returns {<a>}.

5. Elementary relations

If concept a contains intensionally concept b, there is a path form a to b. The Boolean function *contains*(a,b,I-rel) returns true if there is at least one path from a to b.

contains: C-set × C-set × P(C-set × C-set) → {false,true}

contains(a,b,I-rel) =

$$\begin{cases} \text{true, if } \textit{path\_set}(\text{a, b, I - rel}) \neq \{\,\} \\ \text{false, otherwise} \end{cases}$$

The function *is_contained*(a,b,I-rel) is concerned with the inverse relation of intensional containment. The function returns true, if and only if the concept a is intensionally contained in the concept b.

is_contained: C-set × C-set × P(C-set × C-set) → {false,true}

is_contained(a,b,I-rel) =

$$\begin{cases} \text{true, if } \textit{contains}(\text{b, a, C - set, I - rel}) \\ \text{false, otherwise} \end{cases}$$

We get the set of concepts, which are contained intensionally in the concept a by using the function *contains_set*(a,I-rel) in the given relation I-rel.

contains_set: C-set × P(C-set × C-set) → P(C-set)

contains_set(a,I-rel) =

{x ∈ C-set | *contains*(a,x,I-rel)}

For example *contains_set*(clock,I-rel1) yields the set{G, gadget, clock}.

Respectively the function *is_contained_set*(a,I-rel) returns all concepts, whose characteristic concept a is, i.e. the concepts that a is contained in.

is_contained_set: C-set × P(C-set × C-set) → P(C-set)

is_contained_set(a,I-rel) =

{x ∈ C-set | *contains*(x,a ,I-rel)}

In the example *is_contained_set*(clock,I-rel1) yields {clock, clock radio}.

In any non-empty concept system, there is at least one concept, which does not have any other characteristic as itself, and at least one concept, which is not contained in any other concept. Like in [Junkkari and Niinimäki, 1998] the function *genuses* returns the set of the minimal concepts from the concept system i.e. all such concepts, which do not have any successors in the concept relation I-rel.

genuses: $\text{P(C-set} \times \text{C-set)} \rightarrow \text{P(C-set)}$

genuses(I-rel) = *minimal_set*(C-set,I-rel)

Respectively, the function *specie* computes the set of the maximal concepts from concepts system i.e. all concepts that do not have any predecessors in I-rel.

specie: $\text{P(C-set} \times \text{C-set)} \rightarrow \text{P(C-set)}$

specie(I-rel) = *maximal_set*(C-set,I-rel)

Clearly, in the example *genuses*(I-rel1) = {G} and *specie*(I-rel1) = {clock radio}.

6. Association relations

We can associate two concepts a and b, among the third concept c in two ways. Either c contains intensionally both a and b or c is contained in both the concepts a and b. By constituting converses of these situations and by composing all sensible combinations, we get eight Boolean functions altogether. Respectively, we get eight functions that return the sets of concepts, which are in a certain association with concept a.

We say that the concepts a and b are compatible, if and only if there exists a concept which contains both the concepts a and b intensionally. This also means that there is at least one path from a concept x to the concept a and at least one path from x to b. It is easy to see that in I-rel1 all the concepts are compatible with each other.

compatible: $\text{C-set} \times \text{C-set} \times \text{P(C-set} \times \text{C-set)} \rightarrow \{\text{false,true}\}$

compatible(a,b,I-rel) =

$$\begin{cases} \text{true, if } \exists x\text{: } x \in \text{C-set} \wedge \textit{contains}(x,a,\text{I-rel}) \wedge \textit{contains}(x,b,\text{I-rel}) \\ \text{false, otherwise} \end{cases}$$

We say that two concepts a and b are incompatible, if and only if they are not compatible.

incompatible: C-set × C-set × P(C-set × C-set) → {false,true}

incompatible(a,b,I-rel) =

$$\begin{cases} \text{true, if } \neg \mathit{compatible}(\text{a, b, I - rel}) \\ \text{false, otherwise} \end{cases}$$

By using the function compatible we form the set of those concepts, which are compatible with concept a.

compatible_set: C-set × P(C-set × C-set) → P(C-set)

compatible_set (a,I-rel) =

{x ∈ C-set | *compatible*(a,x,I-rel)}

Respectively, we can form the set of concepts incompatible with concept a.

incompatible_set: C-set × P(C-set × C-set) → P(C-set)

incompatible_set(a,I-rel) =

{x ∈ C-set | *incompatible*(a,x,I-rel)}

In general, any C-set is identical with the union of compatible concepts and incompatible concepts of the given concept a. So we could have defined them one by another.

Another basic associate relation is comparability. We say that two concepts a and b are comparable, if and only if there is a concept, which is contained intensionally in both the concepts a and b. In I-rel1 every two concepts are comparable with each other.

comparable: C-set × C-set × P(C-set × C-set) → {false,true}

comparable(a,b,I-rel) =

$$\begin{cases} \text{true,if } \exists \text{x} : \text{x} \in \text{C - set} \wedge \mathit{contains}(\text{a, x, I - rel}) \wedge \mathit{contains}(\text{b, x, I - rel}) \\ \text{false, otherwise} \end{cases}$$

The concepts are incomparable, if and only if they are not comparable.

incomparable: C-set × C-set × P(C-set × C-set) → {false,true}

incomparable(a,b,I-rel) =

$$\begin{cases} \text{true, if } \neg\, \textit{comparable}(\text{a, b, I - rel}) \\ \text{false, otherwise} \end{cases}$$

The *comparable_set*(a,I-rel) contains the concepts, which are comparable with the concept a.

comparable_set: C-set × P(C-set × C-set) → P(C-set)

comparable_set(a,I-rel) =

{x ∈ C-set | *comparable*(a,x,I-rel)}

And respectively the function *comparable_set*(a,I-rel) returns the set of concepts that are incomparable with concept a.

incomparable_set: C-set × P(C-set × C-set) → P(C-set)

incomparable_set(a,I-rel) =

{x ∈ C-set | *incomparable*(a,x,I-rel)}

It holds that C-set is identical with the union of *comparable_set*(a,I-rel) and *incomparable_set*(a,I-rel) for any concept a.

By combination of compatible and comparable concepts, we get homogen-compatible concepts. The function *homogen-compatible*(a,b,I-rel) returns true, if and only if a and b are compatible and comparable. The function *homogen-compatible_set*(a,I-rel) produces the set of concepts, which are homogen-compatible with concept a.

homogen-compatible: C-set × C-set × P(C-set × C-set) → {false,true}

homogen-compatible(a,b,I-rel) =

$$\begin{cases} \text{true, if } \textit{compatible}(\mathrm{a, b, I\text{-}rel}) \ \wedge \ \textit{comparable}(\mathrm{a, b, I\text{-}rel}) \\ \text{false, otherwise} \end{cases}$$

homogen-compatible_set: C-set $\times$ P(C-set $\times$ C-set) $\rightarrow$ P(C-set)

homogen-compatible_set(a,I-rel) =

{x $\in$ C-set | *homogen-compatible*(a,x,I-rel)}

The concepts a and b are *heterogen-compatible* if they are incomparable and compatible. The function *heterogen-compatible_set*(a,I-rel) returns the sets of concepts, which are heterogen-compatible with concept a.

heterogen-compatible: C-set $\times$ C-set $\times$ P(C-set $\times$ C-set) $\rightarrow$ {false,true}

heterogen-compatible(a,b,I-rel) =

$$\begin{cases} \text{true, if } \textit{compatible}(\mathrm{a, b, I\text{-}rel}) \ \wedge \ \textit{incomparable}(\mathrm{a, b, I\text{-}rel}) \\ \text{false, otherwise} \end{cases}$$

heterogen-compatible_set: C-set $\times$ P(C-set $\times$ C-set) $\rightarrow$ P(C-set)

heterogen-compatible_set(a,I-rel) =

{x $\in$ C-set | *heterogen-compatible*(a,x,I-rel)}

We say that two concepts are oppositions (Gegensatz) if they are comparable and incompatible. The function *opposition_set*(a,I-rel) returns the sets of concepts that are opposed to concept a.

opposition: C-set $\times$ C-set $\times$ P(C-set $\times$ C-set) $\rightarrow$ {false,true}

opposition(a,b,I-rel) =

$$\begin{cases} \text{true, if } \textit{incompatible}(\mathrm{a, b, I\text{-}rel}) \ \wedge \ \textit{comparable}(\mathrm{a, b, I\text{-}rel}) \\ \text{false, otherwise} \end{cases}$$

opposition_set: C-set $\times$ P(C-set $\times$ C-set) $\rightarrow$ P(C-set)

opposition_set(a,I-rel) =

{x $\in$ C-set | *opposition*(a,x,I-rel)}

The last associate relation is *isolation* (isoliert). Two concepts a and b are isolated, if they are incomparable and incompatible. The set functions *isolated_set*(a,I-rel) returns the set of concepts, which are isolated with a.

isolated: C-set × C-set × P(C-set × C-set) → {false,true}

isolated(a,b,I-rel) =

$$\begin{cases} \text{true, if } \textit{incompatible}(\text{a, b, I - rel}) \ \wedge \ \textit{incomparable}(\text{a, b, I - rel}) \\ \text{false, otherwise} \end{cases}$$

isolated_set: C-set × P(C-set × C-set) → P(C-set)

isolated_set(a,I-rel) =

{x ∈ C-set | *isolated*(a,x,I-rel)}

Let us take an example. I-rel2 = {<c1,c2>, <c1,c8>, <c2,c6>, <c3,c4>, <c3,c5>, <c4,c6>, <c4,c7>, <c5,c7>, <c5,c8>} is a relation on the concept set {c1, c2, c3, c4, c5, c6, c7, c8} and we consider association relations under it. There are two members in the set *specie*(I-rel2), which are c1 and c3. Respectively, the function *genuses*(I-rel2) returns the set {c6, c7, c8}.

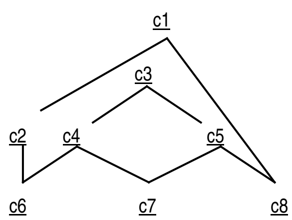

Figure 2. I-rel2

Let us take concept c4 as object for more circumstantial study. First there are three characteristics for c4 and c4 is a characteristic for two concepts. We get the sets of these concepts by using function *contains_set* and the function *is_contained_set*.

contains_set(c4,I-rel2) = {c4, c6, c7}

is_contained_set(c4,I-rel2) = {c3, c4}

There are six concepts that have a common characteristic with the concept c4 and one concept that has no common characteristic with concept c4. The sets of concepts, which are comparable or incomparable with concept c4 are given using the functions *comparable_set*(c4,I-rel2) and *incomparable_set*(c4,I-rel2):

compatible_set (c4,I-rel2) = {c3, c4, c5, c6, c7, c8}
incompatible_set(c4,I-rel2) = {c1, c2}

comparable_set(c4,I-rel2) = {c1, c2, c3, c4, c5, c6, c7}
incomparable_set(c4,I-rel2) = {c8}

The rest of the functions return the following sets:
homogen-compatible_set(c4,I-rel2) = {c3, c4, c5, c6, c7}
heterogen-compatible_set(c4,I-rel2) = {c8}
opposition_set(c4,I-rel2) = {c1, c2}
isolated_set(c4,I-rel2) = { }

The function *isolated_set*(c4,I-rel2) yields the empty set. If the system is not connected in graph theory meaning, there have to be isolated concepts. There are some connected structures that are able to produce isolation between concepts, too.

7. Intensional product and sum

There are several concept operations in Kauppi's concept theory. First we consider intensional sum and intensional product.

According to Kauppi's theory, the intensional product is an unambiguous concept. The theory says that concept x is equivalent with the *intensional product* of the concepts a and b, if and only if x has every common characteristic of a and b, but not any other characteristic except itself. If we assume, like Kauppi did, that there is exactly one intensional product for any comparable pair of concepts in the concept system, we rule out some concept systems that are totally appropriate in everyday modelling. Here, we define intensional product in a more relaxed manner so that concept x inheres in the set of product concepts of a and b if the following condition holds: If x contains concept y intensionally, consequently a and b contain y intensionally. To formulate the function we use functions *contains_set* and *maximal_set*. The concept x satisfies the condition of intensional product if

1. x is a characteristic for both operand concepts a and b,
2. within these characteristics, x is among the maximal concepts.

$$\textit{prod\_set}: \text{C-set} \times \text{C-set} \times \text{P(C-set} \times \text{C-set)} \rightarrow \text{P(C-set)}$$

$$\textit{prod\_set}\ (\text{a,b,I-rel}) =$$

$$\textit{maximal\_set}(\textit{contains\_set}(\text{a,I-rel}) \cap \textit{contains\_set}(\text{b,I-rel}),\text{I-rel})$$

In the case of I-rel1 intensional product between any two concepts is simple. For example, the function *prod_set*(radio,clock,I-rel1) returns the set, whose single member is gadget. By following the function *prod_set*(radio,clock,I-rel1), the intersection of radio's and clock's characteristics is {gadget, G}. The maximal of these is gadget.

Let us take a more complicated situation. The relation I-rel3 = {<c1,c3>, <c1,c4>, <c2,c3>, <c2,c4>, <c4,c5>, <c3,c5>} is the relation on the set C-set3 = {c1, c2, c3, c4, c5}. Now the function *prod_set*(c1,c2,I-rel3) returns the set {c3, c4}.Here the common characteristics of the concepts c1 and c2 are c3, c4 and c5. Among these, c3 and c4 are the maximal ones.

In our last example here we consider a concept system that has a structure like in the figure 3 (I-rel4 = {<c1,c2>, <c1,c3>, <c2,c4>, <c3,c4>, <c4,c5>, <c4,c6>, <c5,c7>, <c6,c7>}). Next we consider the intensional product between concepts c2 and c3 in the relation I-rel4. The intersection of the sets of characteristics (contains_sets) for c2 and c3 is {c4, c5, c6, c7}. The function *prod_set*(c2,c3,I-rel4) returns the set {c4}.

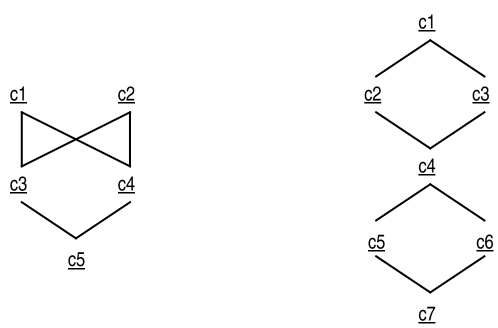

Figure 3. I-rel3 and I-rel4

Intensional sum is considered analogously to intensional product. The theory [Kauppi, 1967] says that concept c is equivalent with intensional sum of the concept a and b, if and only if for every x it holds: c contains a intensionally, if and only if x contains a and b intensionally. Like with intensional product, we define the function, which is more relaxed than the original definition. So a concept x inheres in that set the function *sum_set*(a,b,I-rel) returns, if:

1. x is among the concepts that contain both operand concepts a and b intensionally,
2. x is among the minimal of these.

sum_set: C-set × C-set × P(C-set × C-set) → P(C-set)

sum_set (a,b,I-rel) =

minimal_set(*is_contained_se*t(a,I-rel) ∩ *is_contained_set*(b,I-rel),I-rel)

The function *sum_set* works in the same way as the *prod_set*, but inversely. For example s*um_set*(clock,radio,I-rel1) produces the set {clock radio}. Respectively *sum_set*(c3,c4,I-rel3) returns the set {c1, c2}. In the last example *sum_set*(c5,c6,I-rel4) returns {c4}.

8. Intensional negation and reciprocal

The definition forms of intensional negation and intensional reciprocal correspond to each other. Intensional negation of concept a is the maximal common characteristic for incompatible concepts of a. Intensional reciprocal of concept a is the minimal concept, which contains intensionally all that concepts which are incomparable with a.

According to Kauppi's theory the intensional negation -a of concept a is a concept, which is

- incompatible with a and
- for every x: if a concept x is incompatible with a, then -a is one of the characteristics of x.

If there are not any incompatible concepts with concept b, there cannot be the intensional negation of concept b. In Kauppi's theory, there is an axiom which says that if there exists an incompatible concept with b, there must also exist the intensional negation of the concept b. Therefore, the structure of the concept system is very dense. On account of this, we consider intensional negation in a more spacious way. That is, we consider the negation set instead of a single negation.

In our approach the negation of concept is understood as follows: If a concept b is incompatible with concept a and b has not any such characteristic (except itself) that is incomparable with concept a, then concept b inheres in the *neg_set*(a,I-rel). In the definition of the function, we first consider the set concepts that are incompatible with the concept a and secondly we choose the minimal concepts from this set.

neg_set: C-set × P(C-set × C-set) → P(C-set)

neg_set(a,I-rel) =

minimal_set(*incompatible_set*(a,I-rel),I-rel)

It is worth noticing here that if there is exactly one concept in the *neg_set*(a,I-rel), then this concept corresponds to Kauppi's meaning of intensional negation. Let us take new a relation I-rel5 = {<c1,c4>, <c1,c5>, <c2,c4>, <c2,c6>, <c3,c5>, <c3,c6>, <c4,c7>, <c5,c7>, <c6,c7>} on the concept set {c1, c2, c3, c4, c5, c6, c7}. Now the function results in the different cases are following: *neg_set*(c1,I-rel5) = {c6}, *neg_set*(c2,I-rel5) = {c5}, *neg_set*(c3,I-rel5) = {c4}, *neg_set*(c4,I-rel5) = {c3}, *neg_set*(c5,I-rel5) = {c2}, *neg_set*(c6,I-rel5) = {c1} and *neg_set*(c7,I-rel5) = {}. In this example, if the negation of a concept exists, the concept is equivalent with its negation's negation.

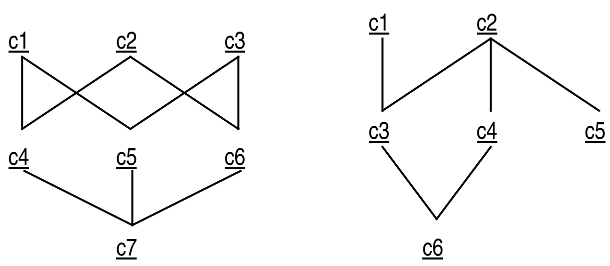

Figure 4. I-rel5 and I-rel6.

In the relation I-rel5 every negation set corresponds to Kauppi's definition for intensional negation, because there is maximally one concept in each negation set. The function *neg_set*(c7,I-rel5) is an empty set, because c7 is compatible with every concept. Let us modify the situation. In I-rel6 = {<c1,c3>, <c2,c3>, <c2,c4>, <c2,c5>, <c3,c6>,

<c4,c6>}, there is not any unequivocal intensional negation for every concept. The concepts c3 and c6 are compatible with every concept. The function *neg_set*(_,I-rel6) with one of the arguments c2, c4 or c5 returns the set {c1}. On the other hand, neg_set(c1,I-rel6) produces the set {c4, c5}. In I-rel6 there are no such concepts which would be equivalent with negation's negation of the concept.

The formulation of intensional reciprocal corresponds with intensional negation. Intensional reciprocal of concept a is the least such concept that contains intensionally all such concepts that are incomparable with concept a.

Like with intensional negation we define analogously the function that produces the set of concepts x, so that x is among the maximal concept incomparable with a.

reciprocal_set: C-set × P(C-set × C-set) → P(C-set)

reciprocal_set (a,I-rel) =

maximal_set(*incomparable_set*(a,I-rel),I-rel)

In the I-rel5, there is not any reciprocal to any concept, because all the concepts are comparable with each other. Instead, the function *reciprocal_set*(_,I-rel6) with one of the arguments c1, c3, c4 or c6 produce the set {c5}. Respectively *reciprocal_set*(c5,I-rel6) returns the set {c1, c4}.

9. Intensional quotient and difference

According to Kauppi's theory, the intensional quotient from concept a to concept b is the concept that is a characteristic for every such concept that contains concept a intensionally and is incompatible with b.

As earlier in this paper, we define the function that produces the set of concepts, which have the most essential lineaments of the original definition. If a concept x (≠a) inheres in this set that the function *quo_set*(a,b,I-rel) produces, it must hold:

1. x contains a intensionally
2. x is incompatible with b
3. x is among the minimal in the set of concepts that satisfy points 1 and 2.

If concept a satisfies the point 2, then the function yields {a}.

quo_set: C-set × C-set × P(C-set × C-set) → P(C-set)

quo_set(a,b,I-rel) = *minimal_set*(S,I-rel)

where S = *is_contained_set*(a,I-rel) ∩ *incompatible_set*(b,I-rel)

Intensional difference from concept a to concept b is the concept that contains intensionally all those concepts that are characteristics for concept a and are incomparable with b.

We define the function *diff_set* that produces the set of concepts, which satisfy the following conditions. A concept x (≠a) inheres in the set *diff_set*(a,b,I-rel) if:

1. a contains x intensionally
2. x is incomparable with b
3. x is among the maximal in the intersection of sets that points 1 and 2 indicate.

If concept a satisfies the point 2., the function yields the set {a}.

diff_set: C-set × C-set × P(C-set × C-set) → P(C-set)

diff_set(a,b,I-rel) = *maximal_set*(S,I-rel)

where S = *contains_set*(a,I-rel) ∩ *incomparable_set*(b,I-rel)

In the example I-rel7 = {<c1,c3>, <c1,c4>, <c2,c4>}, some non-empty *quo_set* sets can be produced. The functions *quo_set*(c2,c1,I-rel7) and *quo_set*(c2,c3,I-rel7) return the set {c2}. The functions *quo_set*(c1,c2,I-rel7) and *quo_set*(c3,c2,I-rel7) return the set {c1}.

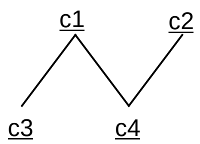

Figure 5. I-rel7

Respectively as above, the function *diff_set*(c2,c3,I-rel7) returns the set {c2} and *diff_set*(c4,c3,I-rel7) returns {c4}. The functions *diff_set*(c3,c2,I-rel7) and *diff_set*(c3,c4,I-rel7) return the set {c3}.

10. Summary

In this paper, we have considered association relations and concept operations of Kauppi's concept theory and presented an abstract implementation of it. Our meta-language here was set theory, which is an established and well-known presentation language.

By using the basic relation, intensional containment, we can make association of two concepts a and b among a third concept c. If c contains both a and b intensionally, we say that the concepts are compatible. If c is contained in both the concepts a and b, we say that concepts are comparable. By constituting these and the converse of these situations and by composing all sensible combinations, we get eight Boolean functions altogether. Respectively, we get eight functions, which return the sets of concepts, which are in a certain association with a given concept.

If we consider the relationships between one or two concepts and their associations to the whole concept system, we talk about concept operations.

Intensional product and sum are relations among three concepts. Intensional product applies to common characteristics and intensional sum applies to concepts that contain both the operand concepts intensionally.

Intensional negation and reciprocal apply to concepts that have a strong divergence from the operand concept. Intensional negation makes reference to those concepts that are incompatible with the operand concept and, respectively, intensional reciprocal refers to those concept, which are incomparable with the operand concept.

Unlike intensional sum and product, intensional difference and quotient are not symmetric relations. In other words, intensional difference or quotient from a to b is not the same as the difference or quotient from b to a. By using the intensional difference from a to b is it possible to derive the concepts, which contain intensionally concept a, but are incomparable with concept b. Intensional quotient, from a to b applies to those concepts that contain concept a intensionally, but are incompatible with concept b.

We have extended the concept theory in two ways. On one hand we have presented operations that return sets of those concepts that are in a considered association with the given concept. On the other hand, we have represented a more spacious interpretation for concept operations. Our functions do not return a single concept, but the set of concepts that satisfy the most essential conditions. Therefore the functions provide a non-empty result, though there is not a unique concept that satisfied the original definition. However, in our view, by using the functions that we have presented it is possible to check also if the structure of a concept system complies with some axiomatic system.

By using a set-oriented approach we have presented in an explicit and well-known manner the formal concept theory, which we see to have potential to manage knowledge and describe the structure of knowledge.

References

[Artale, Franconi and Guarino, 1996] Alessandro Artale, Enrico Franconi and Nicola Guarino, Open problems for part-whole relations, *Proceeding of International Workshop on Description Logics*. Boston MA, November 1996.

[Artale, Franconi, Guarino and Pazzi, 1996] Alessandro Artale, Enrico Franconi, Nicola Guarino and Luca Pazzi, Part-whole relations in object-centered systems: An overview, *Data and knowledge engineering*, Vol. 20, No. 3, North-Holland, Elsevier, 1996.

[Borgida *et al,* 1989] Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness and Lori Alperin Resnick, Classic: A structural data model for objects, *ACM Sigmod record.* Vol. 18, No. 2, 1989. pp. 58-67

[Borgida, 1995] Alexander Borgida, Description logics in data management, *IEEE Transactions on knowledge and data engineering.* Vol. 7, No. 1, 1995. pp. 671-682

[Brachman and Schmolze, 1985] Ronald J. Brachman and J. Schmolze, An overview of the KL-ONE knowledge representation system, *Cognitive science.* Vol. 9 No. 2, 1985. pp. 171-216

[Guarino, 1997] Nicola Guarino, Understanding, building and using ontologies, A Commentary to "Using Explicit Ontologies in KBS Development", by van Heijst, Schreiber, and Wielinga, *International journal of human and computer studies.* Vol. 46, No. 2/3, 1997. pp. 293-310

[Junkkari and Niinimäki, 1998] Marko Junkkari and Marko Niinimäki, An algebraic approach to Kauppi's concept theory, *Proceedings of the 8th European-Japanese Conference on Information Modelling and Knowledge Bases*. May 26-29, Finland, 1998. pp. 115-130

[Järvelin and Niemi, 1993] Kalervo Järvelin and Timo Niemi, Deductive information retrieval based on classifications, *Journal of the American society for information science*. Vol. 44, Number 10, New York, 1993.

[Kangassalo, 1993] Hannu Kangassalo, COMIC: A system and methodology for conceptual modelling and information construction, *Data and knowledge engineering 9*. Elsevier Science Publishers, North-Holland, 1993. pp. 287-319

[Kangassalo, 1996] Hannu Kangassalo, Conceptual description for information modelling based on intensional containment relation, in Franz Baader, Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt (Eds.): *Knowledge Representation Meets Databases*, Proceedings of the 3rd Workshop KRDB'96, Budapest, Hungary, August 13, 1996.

[Kauppi, 1967] Raili Kauppi, *Einführung in die Theorie der Begriffssysteme*. Acta Universitatis Tamperensis, Ser. A, Vol. 15, Tampereen yliopisto, Tampere, 1967.

[Lambrix, 1996] Patrick Lambrix, *Part-Whole Reasoning in Description Logics*, Linköping Studies in Science and Technology, Dissertation No. 448, Linköping, 1996.

[Niemi and Järvelin, 1992] Timo Niemi and Kalervo Järvelin, Operation-oriented query language approach for recursive queries - Part 1: Functional definition, *Information systems*. Vol. 17 No. 1, Pergamon Press plc, 1992. pp. 49-75

[Smith and Smith, 1977] J. M. Smith and D. C. Smith, Database abstraction: Aggregation and generalization, *ACM Transactions on database systems.* Vol. 2, No. 2, 1977. pp. 105-133

[Sowa, 1984] John F. Sowa, *Conceptual Structures*. Addison Wesley, 1984.

[Woods, 1991] W. A. Woods, Understanding subsumption and taxonomy: A framework for progress, in J. F. Sowa (ed.), *Principles of Semantic Networks.* Morgan Kaufman Publishers, San Mateo, California, 1991. pp. 45-94

Appendix: The definitions and operations of this paper as presented by Kauppi

Comparable:	Df_{H} ”$a\ H\ b$” $=_{df}$ ”$(\exists x)(a \geq x \ \&\ b \geq x)$”
Incomparable:	$Df_{\text{⌶}}$ ”$a \mathrel{\text{⌶}} b$” $=_{df}$ ”$\sim(\exists x)(a \geq x \ \&\ b \geq x)$”
Compatible:	$Df_{\text{⅄}}$ ”$a \mathrel{\text{⅄}} b$” $=_{df}$ ”$(\exists x)(x \geq a \ \&\ x \geq b)$”
Incompatible:	Df_{Y} ”$a\ Y\ b$” $=_{df}$ ”$\sim(\exists x)(x \geq a \ \&\ x \geq b)$”
Homogen-compatible:	$Df_{\underline{\text{⅄}}}$ ”$a \mathrel{\underline{\text{⅄}}} b$” $=_{df}$ ”$a\ H\ b \ \&\ a \mathrel{\text{⅄}} b \ \&\ \sim a \geq b \ \&\ \sim b \geq a$”
Heterogen-compatible:	$Df_{\text{⅄}\!\!\perp}$ ”$a \mathrel{\text{⅄}\!\!\perp} b$” $=_{df}$ ”$a \mathrel{\text{⌶}} b \ \&\ a \mathrel{\text{⅄}} b$”
Opposite:	$Df_{\overline{Y}}$ ”$a \mathrel{\overline{Y}} b$” $=_{df}$ ”$a\ H\ b \ \&\ a\ Y\ b$”
Isolated:	$Df_{\top\!\!\!\!Y}$ ”$a \mathrel{\top\!\!\!\!Y} b$” $=_{df}$ ”$a \mathrel{\text{⌶}} b \ \&\ a\ Y\ b$”
Sum:	$Df_{\oplus}$ ”$c = a \oplus b$” $=_{df}$ ”$(x)(x \geq c \leftrightarrow x \geq a \ \&\ x \geq b)$”
Product:	$Df_{\otimes}$ ”$c = a \otimes b$” $=_{df}$ ”$(x)(c \geq x \leftrightarrow a \geq x \ \&\ b \geq x)$”
Negation:	$Df_{\_}$ ”$b = \bar{a}$” $=_{df}$ ”$(x)(x \geq b \leftrightarrow x\ Y\ a)$”
Difference:	$Df_{\ominus}$ ”$c = a \ominus b$” $=_{df}$ ”$(x)(c \geq x \leftrightarrow a \geq x \ \&\ b \mathrel{\text{⌶}} x)$”
Quotient:	$Df_{\oslash}$ ”$c = a \oslash b$” $=_{df}$ ”$(x)(x \geq c \leftrightarrow x \geq a \ \&\ x\ Y\ b)$”
Reciprocal:	Df_{R} ”$b = \ominus a$” $=_{df}$ ”$(x)(b \geq x \leftrightarrow a \mathrel{\text{⌶}} x)$”

Notes:

- There is a printing error in [Kauppi, 1967, p. 44] in the definition of opposite.
- Logical notions are following: logical negation (~), logical conjunction (&), universal quantifier ((x) means ($\forall$x)) .
- The symbol ”$\geq$” is used here instead of Kauppi’s ”>”.
- Kauppi defines the reciprocal only in calculus BK*, where greek symbols are used for concept. Here, they are represented in latin letters.
- Please observe the difference between the definition of homogen-compatible in Kauppi’s work and in this paper.