A Semiotic-Conceptual Analysis of Conceptual Learning*

Uta Priss

Zentrum für erfolgreiches Lehren und Lernen
Ostfalia University of Applied Sciences
Wolfenbüttel, Germany
www.upriss.org.uk

Abstract. While learning mathematics or computer science, beginning students often encounter significant problems with abstract concepts. In both subjects there tend to be large numbers of students failing the class or dropping out during the first semesters. There is a substantial existing body of literature on this topic from a didactic perspective, but in our opinion an investigation from a semiotic-conceptual perspective could provide further insights and specifically analyse the difficulties encountered when learning abstract concepts. This means that both the complexities of the representations of abstract concepts and the conceptual content itself are modelled and investigated separately and in combination with each other. In our opinion a semiotic analysis of the representations is often missing from didactic theories. And in particular, as far as we know, there are not yet any formal mathematical approaches to modelling learning difficulties with respect to semiotic and conceptual structures. Semiotic-Conceptual Analysis (SCA) as presented in this paper aims to fill that niche.

1 Introduction

Semiotic-Conceptual Analysis (SCA) was inspired by Charles S. Peirce's triadic definition of signs but does not claim to present an exact formalisation of his ideas. A more detailed discussion of how SCA relates to Peirce was provided by Priss (2015) and shall not be repeated in this paper. Peirce's semiotics was aimed at analysing signs occurring in natural communication where representamens (physical representations of signs) are visible or audible (in form of words, gestures and so on) but denotations (meanings) and mental interpretations can only be speculated about. Nowadays computer programs are examples of sign communication where every aspect of the signs, their representations, inputs, outputs, states and runtime behaviour can be documented in minute detail. Furthermore modern programming languages display a variety of complex structures (such as abstract data types, object orientation or functions as first class objects) which are probably far beyond the complexity that is expressible within natural languages. Thus formal languages are an interesting domain for semiotic analyses. Analysing program source code was one of the motivations for developing SCA (Priss, 2015). Another interesting semiotic aspect is how people interact with such formal representations and,

* Published in Haemmerle et al, Proc of ICCS'16, LNAI 9717

also, what difficulties students encounter when they are learning to interpret and use formalisms in mathematics and computer science. That is the focus of this paper. It shows how SCA can serve as a tool for exploring and highlighting difficulties within representations and their underlying abstract content. While there are already many existing approaches to semiotics, our goal is to develop a theory that builds on mathematical formalisations of signs and of concepts in the sense of Formal Concept Analysis (FCA[1]). To our knowledge such an FCA-based theory of semiotics does not yet exist elsewhere.

SCA defines signs as instances of a triadic relation consisting of representamens, denotations (or meanings) and interpretations. Interpretations are functions that map representamens into denotations. Peirce uses the term 'object' instead of 'denotation' and 'interpretant' instead of 'interpretation'. Priss (2015) explains why SCA adopts a different terminology. It should be emphasised that similar to how FCA uses its notions, the terms 'representamen', 'denotation' and 'interpretation' indicate structural positions within the formal model presented by SCA. As will be explained below such notions are 'anonymous signs' in the terminology of SCA. When explaining FCA to non-mathematicians one sometimes encounters criticisms such as 'what you are calling concepts are not concepts'. But from a mathematical viewpoint, 'concept' is just a name for a structure. It does not mean anything other than what is defined. Only in applications such notions acquire additional meaning which can be investigated with respect to their appropriateness in other domains. For example, whether or not SCA provides a 'semiotic analysis' in the ordinary sense depends on how it is used in an application. From a teaching perspective, the use of anonymous signs may be one of the core difficulties that students encounter when they learn mathematics. Students often associate concrete meanings with anonymous signs or fixate on specific representations instead of realising that the meaning of the sign in question is purely structural.

Our goal for SCA is to describe a semiotic theory that is applicable to all signs and all kinds of representamens. Each of the three components of signs (representamens, denotations and interpretations) has conceptual structures and some form of similarity. For SCA we use concept lattices but the core approach and terminology of SCA would still be applicable even if, for example, conceptual graphs were used instead of concept lattices. One of the core questions is whether similar representamens have similar denotations under similar interpretations. From an educational viewpoint one can investigate, for example, whether the interpretations used by a student are similar to the interpretations used by a teacher. If a student has 'understood' a concept then he or she should use the signs relating to this concept in a similar manner as a teacher.

The next section explains briefly why it is useful to define different notions of similarity for signs (such as synonymy) instead of just defining equality. Section 3 of this paper provides a brief overview over other existing theories that are relevant in this context. Section 4 establishes the formal definitions of SCA. Sections 5 and 6 demonstrate how SCA can be used for didactic applications. The paper finishes with a concluding section.

---

[1] Because Formal Concept Analysis (FCA) has been presented many times at this conference, this paper does not provide an introduction to FCA but there is an example with some explanation in Section 5. Further information about FCA can be found, for example, on-line (http://www.upriss.org.uk/fca/) and in the main FCA textbook by Ganter & Wille (1999).

2 Equality and similarity of signs

One of the challenges for semiotics is that it is common to ignore certain aspects of signs in everyday language where a sign and its representamen are not always clearly distinguished. In mathematics, representamens tend to be ignored and equality tends to be denotational. For example, $x = 5$ means that the value of x is 5. Obviously the representamens, a letter x and a number 5, are different. This raises the question as to what it means for two signs to be equal to each other. It should be mentioned that a sign can be observed at different levels of granularity corresponding to a sentence, word or character. For example if someone expresses the sentence "I just bought an apple" on two consecutive days then it is a question whether this constitutes a single sign or two different signs. Most likely two different apples will be involved. Nevertheless, as elaborated by Priss (1998) the meaning of a word is not an object (a real apple) but a concept ('apple'). This concept could still be the same on both days. Because signs are triadic, however, two signs can only be equal if their representamens, denotations and interpretations are all equal (or at least equivalent). If the interpretations[2] contain information about the spatial and temporal context in which the sign was used, then the two sentences about eating an apple must be different signs. If the interpretations are less detailed, then it could be the same sign used on different days. We use the term 'equinymy' to describe this case where signs have equivalent representamens and equal denotations but possibly different interpretations. How interpretations are chosen is up to the person who uses SCA for modelling. Thus whether or not the sentence above corresponds to one sign or to two equinymous signs is a consequence of modelling decisions. If on the first day 'apple' refers to fruit and on the second day to a computer, then two different interpretations (and thus two signs) are required because interpretations are functions. In that case the term 'homograph' describes signs that have the same representamen but totally different meanings.

Synonymous signs have different representamens but similar meaning. Tab. 1 shows four pairs of representamens with similar meanings. Any statements that can be made about these always depend on interpretations. The first two representamens are distinguished by their font. SCA uses tolerance or equivalence relations[3] which express which representamens are considered to be similar or equivalent. If the representamens in the first row are considered equivalent, then they belong to the same sign or to equinyms depending on whether one or two interpretations are involved. The representamens in the second row are already less similar to each other. $T\sigma\alpha\rho\lambda\varsigma\ \Sigma.\ \Pi\varepsilon\rho\varsigma$ is the modern Greek spelling of Peirce's name. The number of characters in both representamens is different, thus it might be difficult to establish representamen similarity by a simple mapping of characters. Most likely two interpretations are required in this case. It is even more difficult to detect similarity for the third pair. The 24 numbers for the hours of the right clock correspond to twice the number of hour-lines of the left clock. The minutes of the right clock correspond to the movement of the minute hand of the left

[2] 'Interpretation' in SCA is an anonymous sign and refers to a function. 'Having a different interpretation' in SCA means using a different function. Different interpretations can still lead to the same denotation. This is different to how 'interpretation' is used in ordinary language.

[3] A tolerance relation is symmetric and reflexive. An equivalence relation is also transitive.

clock. There is a representation of movement in both clocks. Thus several aspects of similarity can be established. Finally, the two representamens in the last row only have in common that they are both sets. Otherwise there is no similarity between their representamens. They are still synonyms if their interpretations map them onto the same denotation.

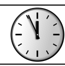
1	CHARLES S. PEIRCE	*Charles S. Peirce*	equal signs or equinyms
2	$T\sigma\alpha\rho\lambda\varsigma\ \Sigma.\ \Pi\varepsilon\rho\varsigma$	Charles S. Peirce	(equal signs,) equinyms or strong synonyms
3		23:56	(equal signs,) equinyms or strong synonyms
4	$\{1, 2, 3\}$	$\{n \in N \mid 1 \leq n \leq 3\}$	strong synonyms

Table 1. Different representamens with similar meanings

3 How SCA relates to other existing theories

The majority of other existing semiotic theories appear to be either not formally (mathematically) defined or not triadic. Priss (2015) briefly discusses other formalisations of Peirce's semiotics and approaches to model triadic relations with FCA and argues that these are quite different from SCA. Goguen (1999, p. 1) remarks "Semiotics ... much of the research in this area has been rather vague." His own work is not vague but a formal theory of what he calls Algebraic Semiotics. One difference between SCA and Algebraic Semiotics is that although Goguen discusses some of Peirce's ideas, his main influence was Saussure. Thus his signs are binary and belong to sign systems. His definition of sign systems uses partial orders for sorts and constructors. This is somewhat similar to our use of concept lattices as described below. Goguen then discusses structure preserving morphisms among sign systems which are similar to some of the mappings discussed by Priss (2015) and used in sections 5 and 6 of this paper. But another difference between Goguen's work and SCA is that his morphisms are mainly focussed on representamens (in our terminology) and syntactic constructions. We would argue that Goguen's method is very useful, but mainly for representamens that are structurally reasonably similar to each other and not for those that are very different. For example, it is fairly straightforward to construct morphisms for the first three rows in Tab. 1. But for the last row, the only connection between the left and right representamens can be established via their denotations.

Another area that should be mentioned is formal model-theoretic semantics which maps representations into models using interpretations and thus has similar ingredients as SCA. But we would argue that formal semantics is a binary view, and not a triadic one, because their interpretations do not have any structure themselves other than being functions. As far as we are aware, formal semantics is not concerned with questions about the ordering, similarity or quality of interpretations. Formal semantics is mainly concerned with formal languages whereas SCA can be applied to non-formal languages and to questions about how formal languages are used by people as well. In formal semantics it is not possible to discuss the representational and the denotational aspects of a sign separately.

Last but not least, it should be mentioned that semiotics (and SCA) is not the same as usability modelling. While it is possible to ask semiotic questions about how people use signs, one can at the same time ask questions about why certain signs might be used in a certain way based on an analysis of their parts, structures and relations with other signs. Thus usability, semantic and syntactic questions can all be discussed within a single framework in SCA.

4 The core definitions of SCA

This section presents the core mathematical definitions of SCA[4].

Definition 1: A *semiotic relation* $S \subseteq I \times R \times D$ is a relation between three sets (a set R of *representamens*, a set D of *denotations* and a set I of *interpretations*) with the condition that any $i \in I$ is a partial function $i : R \nrightarrow D$. A relation instance (i, r, d) with $i(r) = d$ is called a *sign*.

Alternatively, S can be called a set of signs. The sets R, I, D and S need not be disjoint. Thus a denotation, representamen or interpretation can also be a sign itself. It is possible to use total functions instead of partial functions by adding a NULL-element as shown in the next definition. Formalisations involving a NULL-element can be complex because NULL might correspond to negative, missing or contradictory information. A common programming practice is to deliberately check whether a variable is non-NULL before performing an operation that would otherwise crash and to ignore the problem otherwise. Similarly, we will only mention NULL-elements and the fact that the functions are partial in the text below if it is absolutely necessary.

Definition 2: For a semiotic relation S, a *NULL-element* $d_\perp$ is a special kind of denotation with the following conditions: (i) $i(r)$ undefined in $D \Rightarrow i(r) := d_\perp$ in $D \cup \{d_\perp\}$. (ii) $d_\perp \in D \Rightarrow$ all i are total functions.

The following definitions determine the basic structures for each of the three sets. In each case a concept lattice and tolerance relations are defined. Linguists sometimes use the term ‘open set’ for a set that is large and indeterminate, such as the set of all the words of a natural language. It is feasible to define a tolerance relation on such an open set based on rules. Concept lattices, however, require an explicit set of formal objects. It is therefore advantageous not to incorporate R, I and D directly into concept lattices but to map these sets into concept lattices which model domains for the sets. In applications such mappings might be just partial functions but in that case the sets can be reduced in order to have total functions. Domain lattices can be generated from data or can be preconstructed based on assumptions about the data and then be reused for different applications. Building lattices and defining tolerance relations constitutes by itself some form of interpretation. This can be modelled with SCA as well but that is not further discussed in this paper.

[4] A reader who is unfamiliar with FCA could read Section 5 first because it contains an example of a concept lattice.

Definition 3: For a set R of representamens, a set $T_R = \{t \mid t \subseteq R \times R\}$ of tolerance relations is defined with a subset $E_R \subseteq T_R$ of equivalence relations. A concept lattice $B(O_R, A_R, J_R)$ called *representamen domain lattice* is defined with sets O_R and A_R, a binary relation $J_R \subseteq O_R \times A_R$ and a function $\beta : R \to B(O_R, A_R, J_R)$ with the condition $\exists_{e \in E_R} \forall_{r_1, r_2 \in R} : (r_1, r_2) \in e \Longleftrightarrow \beta(r_1) = \beta(r_2)$.

In applications the condition about mapping equivalent representamens onto the same concept can always be achieved by first constructing β and the lattice and then defining the equivalence relation accordingly. For each tolerance relation on R the function β induces a tolerance relation on the lattice. Ideally a tolerance relation on a lattice should be somehow related to the lattice structure (for example by defining a distance metric on the lattice) but that is a modelling aspect which is not a formal requirement. It should be noted that tolerance relations are expected to be defined on the whole set (i.e., $\forall_{r \in R} : (r, r) \in t$) not just on a subset of R. The next two definitions establish domain lattices for the other sets in a similar manner.

Definition 4: For a set I of interpretations, a set $T_I = \{t \mid t \subseteq I \times I\}$ of tolerance relations is defined with a subset $E_I \subseteq T_I$ of equivalence relations. A concept lattice $B(O_I, A_I, J_I)$ called *interpretation domain lattice* is defined with sets O_I and A_I, a binary relation $J_I \subseteq O_I \times A_I$ and a function $\beta : I \to B(O_I, A_I, J_I)$ with the condition $\exists_{e \in E_I} \forall_{i_1, i_2 \in I} : (i_1, i_2) \in e \Longleftrightarrow \beta(i_1) = \beta(i_2)$.

One possibility for defining the lattice is to choose $O_I = I$ and to define $\beta(i)$ as the lowest concept that contains i in its extension. The interpretations could represent who is interpreting (a native speaker, a teacher or a student, a programming language compiler, and so on) and when and where a sign is used. This could involve a containment hierarchy. For example, there could be an interpretation for a whole book, with separate interpretations for chapters and paragraphs. It is possible to combine all of these containment orders into one concept lattice because using the method of Dedekind closure any partial order can be embedded into a lattice. Once such a lattice has been formed, one can then set the set A_I to correspond to the meet-irreducible lattice elements. But this is just one possibility for constructing the lattice. It could also be constructed in a totally different manner for other applications.

Definition 5: For a set D of denotations, a set $T_D = \{t \mid t \subseteq D \times D\}$ of tolerance relations is defined with a subset $E_D \subseteq T_D$ of equivalence relations with the condition $(d, d_\perp) \in e \in E_D \Rightarrow d = d_\perp$. A concept lattice $B(O_D, A_D, J_D)$ called *denotation domain lattice* is defined with sets O_D and A_D, a binary relation $J_D \subseteq O_D \times A_D$ and a function $\beta : D \to B(O_D, A_D, J_D)$ with the conditions that if $\beta(d_\perp)$ exists it is the bottom element of the lattice and $\exists_{e \in E_D} \forall_{d_1, d_2 \in D} : (d_1, d_2) \in e \Longleftrightarrow \beta(d_1) = \beta(d_2)$.

A denotation domain lattice represents the denotational knowledge of a domain. It could be derived from data or from an ontology (or textbook knowledge in an educational application) using any of the usual FCA techniques for encoding knowledge. The denotational knowledge could also be provided using other knowledge representation techniques (such as conceptual graphs, description logic or formal ontologies). But for the purposes of SCA, one would then need to extract lattices from such knowledge. The next definition shows how some common linguistic terms are formalised in SCA. These definitions only use the relations from Defs. 3-5, not the concept lattices. Thus they would still be applicable if some formalisation other than lattices was used.

Definition 6: For a semiotic relation S with $t \in T_D$, $I_1 \subseteq I$, $e \in E_R$, $e_D \in E_D$,

a) I_1 is *e-compatible* $\Leftrightarrow \forall_{(r_1,r_2)\in e, i_1,i_2 \in I_1, i_1(r_1)\neq d_\perp, i_2(r_2)\neq d_\perp} : (i_1(r_1), i_2(r_2)) \in t$
b) I_1 is *e-mergeable* $\Leftrightarrow \forall_{(r_1,r_2)\in e, i_1,i_2 \in I_1, i_1(r_1)\neq d_\perp, i_2(r_2)\neq d_\perp} : i_1(r_1) = i_2(r_2)$
c) $(i_1, r_1, d_1) = (i_2, r_2, d_2) \Leftrightarrow i_1 = i_2, (r_1, r_2) \in e, d_1 = d_2$.
d) (i_1, r_1, d_1) and (i_2, r_2, d_2) are *strong synonyms* $\Leftrightarrow (r_1, r_2) \notin e$ and $(d_1, d_2) \in e_D$
e) (i_1, r_1, d_1) and (i_2, r_2, d_2) are *equinyms* $\Leftrightarrow (r_1, r_2) \in e$ and $(d_1, d_2) \in e_D$
f) (i_1, r_1, d_1) and (i_2, r_2, d_2) are *synonyms* $\Leftrightarrow (r_1, r_2) \notin e$ and $(d_1, d_2) \in t$
g) (i_1, r_1, d_1) and (i_2, r_2, d_2) are *polysemous* $\Leftrightarrow (r_1, r_2) \in e$ and $(d_1, d_2) \in t$
h) (i_1, r_1, d_1) and (i_2, r_2, d_2) are *homographs* $\Leftrightarrow (r_1, r_2) \in e$ and $(d_1, d_2) \notin t$
i) (i, r, d) is *anonymous* $\Leftrightarrow r = d$

If e is clear from the context, the prefix 'e-' can be omitted. The notions in d) to h) depend on i_1 and i_2, thus in cases of possible ambiguity, one could write 'i_1, i_2-synonyms' and so on. Mergeability means that the interpretations can be merged because the result of the merger is still a function. Representamens as physical manifestations usually display minute variations. For example, two spoken words or two handwritten words are probably never totally equal. Even two computerised images that look the same may not be totally equal if one of them has been compressed or encoded differently. Therefore we allow for two equal signs to have equivalent instead of equal representamens. If $i \in I$ is not e-mergeable with itself, either e could be changed by reducing the size of the equivalence classes or I could be changed by splitting interpretations which are not e-mergeable with themselves.

As mentioned before, it is quite restrictive to require two equal signs to have the same interpretation. Therefore the other notions from Def. 6 describe forms of similarity among interpretations and among signs which are weaker than equality. Synonymy, polysemy and homographs are formalisations of the usual linguistic notions. Equinymy was coined by Priss (2004) and refers to the same representamen being used with the same meaning under different interpretations. Equinymy probably expresses what one might intuitively think it means for signs to be 'the same'. An example of anonymous signs are literals in programming languages. Priss (2004) argues that mathematical variables are anonymous signs as elaborated below.

Proposition 1: For a semiotic relation S with $t \in T_D$, $e \in E_R$, $e_D \in E_D$, $i_1, i_2 \in I$

a) e-compatibility and e-mergeability are tolerance relations on $I \times I$. If $d_\perp$ does not exist, then e-mergeability is an equivalence relation on $I \times I$.
b) For given, fixed i_1 and i_2, equinymy is an equivalence relation, synonymy, strong synonymy, polysemy and homographs are tolerance relations on $S \times S$.
c) Strong synonymy implies synonymy. Equinymy implies polysemy.
d) Compatible interpretations are free of homographs.
e) For mergeable interpretations, polysemy and equinymy are the same.
f) Two anonymous signs are equinyms if their denotations are equal. If $e = e_D$ or e_D is the identity relation, two anonymous signs cannot be strong synonyms. If $t = e$ or t is the identity relation, they cannot be synonyms.
g) If $t = e = e_D$ or t and e_D are identity relations, then two anonymous signs in mergeable interpretations can only be equal (if $i_1 = i_2$), equinyms (if $i_1 \neq i_2$) or not equal. They cannot be synonyms or homographs.

The statements in Prop. 1 follow directly from Def. 6. It should be noted that compatibility and mergeability need not be elements of T_I. The set T_I contains those tolerance relations which are explicitly defined for I, not necessarily those which are emerging based on other defined structures. Statement g) describes how variables are normally used in mathematics. Mathematical texts usually only involve one interpretation which means signs are either equal or not equal. In mathematics, variable names are not distinguishable from their content and equivalence among variable names corresponds to denotational equality. In programming languages, however, signs are not anonymous and interpretations are not always mergeable. We conclude this section with the definition of SCA:

Definition 7: A semiotic relation with concept lattices as presented in Definitions 3-5 is called a *semiotic system*. The study of semiotic systems is called a *semiotic-conceptual analysis*.

A next step is to investigate functions between the concept lattices. Because the interpretations are partial functions from representamens into denotations this leads to the question as to whether they induce partial functions from the representamen domain lattice to the denotation domain lattice. Priss (2015) discusses some conditions for such functions which we will omit in this paper because we believe that further applications are needed in order to determine what is most promising. The following sections show some examples of using such functions in educational applications.

5 An example: reading Hasse diagrams of concept lattices

As a first example we are investigating challenges involved in teaching students about concept lattices and their representations as Hasse diagrams. Fig. 1 shows a formal context and a Hasse diagram of a concept lattice. A Hasse diagram is a representation of a partially ordered set which has the elements as nodes (here nodes 0, 1, ... 4) and their immediate relationships as edges. The edges in the diagram are directed. This means that going up in the diagram corresponds to going up in the partially ordered set. Because the ordering in a partially ordered set is transitive, node 4 is not only below node 3 (its immediate neighbour) but also below all the other nodes. Node 0 is above all other nodes. Transitivity is implied but not explicitly represented in a Hasse diagram because there are no lines between node 4 and node 2 and so on. The implied transitivity needs to be explained to someone who does not know what a Hasse diagram is.

The Hasse diagram in Fig. 1 shows a concept lattice which means that every set of nodes has a unique supremum and a unique infimum and it corresponds to a formal context as displayed in the left-hand side of the figure. The objects are written slightly below the nodes (or concepts) they belong to and the attributes slightly above. In this example the objects are bird, cat and bat and the attributes are flies, mammal and has gills. A concept in FCA has an extension and an intension. An extension consists of all objects from all concepts below a concept. An intension is defined analogously. In this example the concept of node 3 has bat in its extension and flies and mammal in its intension. The theories of lattices in general and of FCA in specific are well developed. Therefore the body of knowledge that is connected with this small lattice is larger than

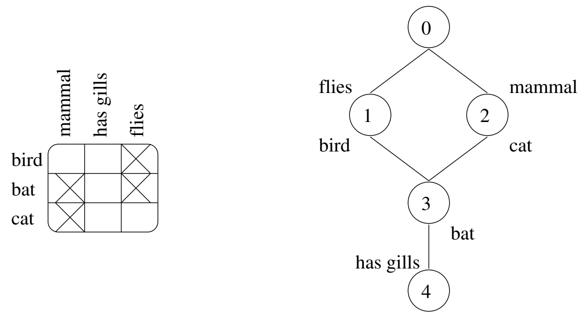


Fig. 1. A formal context and a Hasse diagram of a concept lattice

one might think when one first sees this little figure which consists of just a few nodes, edges and labels.

In our experience with showing FCA to students (and users in general) there are a number of typical questions that arise when they first see Hasse diagrams:

- What is the purpose of the top and bottom node?
- Why are there unlabelled nodes?
- How can the extensions and intensions be read from the diagram?
- What is the relationship between nodes that do not have an edge between them but can be reached via a path?
- What is a supremum or an infimum?
- How can one tell whether it is a lattice?

Fig. 2 shows a modelling of Hasse diagrams with respect to a representamen domain lattice (on the left) and of notions from lattice theory as a denotation domain lattice on the right. In both lattices only the representamens and denotations are shown but not the formal objects and attributes. For the representamen lattice, it is assumed that for a representamen r there is a set $r^{J_R} \subseteq A_R$ of attributes which is assigned to r. The set r^{J_R} need not be an intension of $B(O_R, A_R, J_R)$. Then $\beta(r)$ is defined as the largest concept that contains r^{J_R} in its intension. The definition of $\beta(d)$ is analogously. Building representamen and denotation domain lattices involves modelling. Thus the lattices in Fig. 2 are not to be understood as 'ultimate truth' but instead as a teacher's model. The goal of this is example is not to discuss whether these lattices are correct or not but whether building and analysing such lattices can provide insights about a semiotic relation.

The superconcept ordering in both lattices corresponds to prerequisite knowledge for a student. For example one needs to know what operators and sets are before one can learn what a relation is. The dashed lines in Fig. 2 show the result of an interpretation for some of the representamens. Every concept of the denotation domain lattice represents a different meaning. Because this example comes from a mathematical domain, it may be sufficient to assume a single interpretation. Fig. 2 shows that the structures between the representamen and denotation domain lattices are quite different. In this example

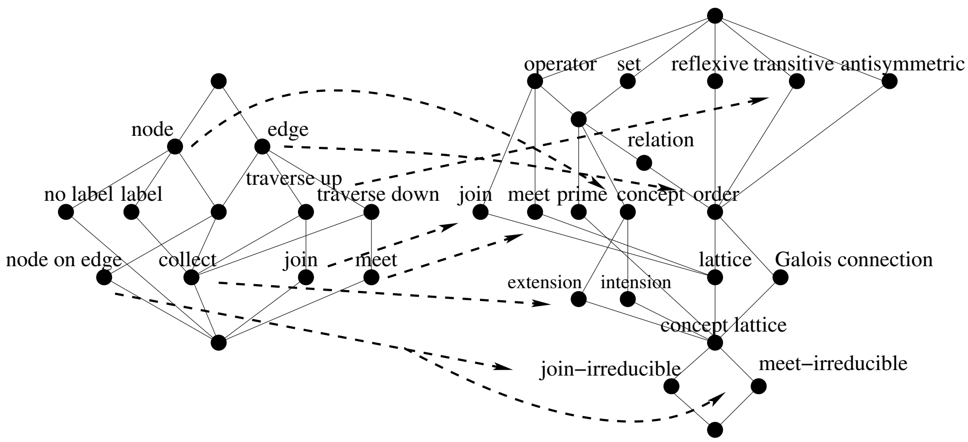


Fig. 2. A representamen and a denotation domain lattice

this means that Hasse diagrams are structurally quite different from lattices. There are connections between some of the core notions, such as 'node' representing 'concept' and 'edge' representing immediate neighbours in the ordering relation. Some of the representamens represent activities on the Hasse diagram: traversing edges up or down and collecting labels on the way represents extensions and intensions. The node/edge relationship is used for identifying irreducible concepts by counting the number of edges that lead to a node from above or from below. Representamens such as labels for objects and attributes do not seem important for the denotation domain lattice, presumably because mathematically they are just elements of sets. The relationships between 'edge' and 'traversal' and 'transitive' and 'order' are inverse. For diagrams one must first know what an edge is before one can talk about traversing edges. But in order to define an order relation one must first know what transitivity is.

In order to address the questions students have when they first see Hasse diagrams of concept lattices, a diagram such as Fig. 2 helps to investigate where exactly their misconceptions are coming from. For example, a question about the purpose of the top and bottom node indicates that the student does not know what a complete lattice is. Using the denotation domain lattice in Fig. 2 a teacher could ask a student about order, join and meet because these are prerequisite to understanding lattices. If the student also has problems with these concepts, then the teacher could move up further in the lattice. It should be mentioned that this approach of representing an ordered set of prerequisite knowledge is also used in Knowledge Space Theory (Albert & Lukas, 1999; Falmagne et al. (2013)). Connections between Knowledge Space Theory and FCA are well known but are beyond the scope of this paper.

6 An example: the meaning of the equality sign

Prediger (2010) discusses how students successively enhance their understanding of the equality sign. At first in primary school, students interpret the equals sign as a request to calculate something (such as '$2 + 3 = ?$') . Prediger calls this the operational use because it is a request for performing an operation. Later, students learn a 'relational' meaning of the equals sign which could involve symmetric identities ($4 + 5 = 5 + 4$), general equivalences ($(a - b)(a + b) = a^2 - b^2$), searching for unknowns ($x^2 = 6 - x$) and contextual uses ($a^2 + b^2 = c^2$) where the variables are meaningful in a context, such as characterising a right-angled triangle. A further, different case are specification uses (such as defining $x := 4$).

Priss et al. (2012) model the denotational content of the equals sign and other equation, assignment and comparison operators using FCA. Their resulting concept lattice is presented in the right-hand side of Fig. 3. We are now revisiting this example by building a representamen domain lattice (left-hand side of Fig. 3) and investigating interpretation-induced partial functions. The left lattice is constructed with $O_R = R$ and $\beta(r)$ is the object concept of r. For the right lattice, $\beta(d)$ is constructed as in the previous example. The denotation domain lattice is built using formal objects which are examples of uses of the equals sign, inequality ($>$), equivalence ($\Leftrightarrow$), basic operations from programming languages: not-equal ($!=$), test for equality ($==$) and Boolean operators (&&). It should be noted that for the examples with more than one operator, the main operator ($\Leftrightarrow$, $==$ and &&) is the one that is investigated. The formal attributes are 'operation', 'contextual', 'definition' (i.e., specification) and 'law' (i.e., equivalence), 'test' and whether the statements are true for all values of the variables or just for some. A 'definition' for other symbols than '$=$' defines a set of possible values for a variable (e.g., $i > 1$). A 'test' is a request to evaluate an expression with respect to variables with given values. The representamen domain lattice classifies the different operation symbols with respect to their parts and their complexity. Even though '$\Leftrightarrow$' could be one character, we argue that one could see it as two arrows and thus as more complex. As for the previous example, the lattices are just an example of modelling by one teacher for the purpose of evaluating student progress.

The dashed lines in Fig. 3 show the result of an interpretation which maps representamens into denotations. A single representamen can be mapped onto the concept in the denotation domain lattice that is the join concept of all its different denotations or a concept of the representamen lattice can be mapped depending on its extension. If a representamen is used for different denotations this means that different interpretations have to be involved. In this example, only $==$ and && are used for exactly one denotation ('test'). Furthermore, the representamens in the extension of the concept labelled 'colon' are only used with one denotation ('definition'). These interpretation instances are shown by the dashed lines. All other representamens and all other extensions of representamens are used for different denotations whose join is the top concept in the denotation domain lattice. These interpretation instances are not drawn in Fig. 3. Overall there is not much structure from the representamen domain lattice that is preserved in the denotation domain lattice. Many representamens are homographs because the join of their images under different interpretations is the top concept of the denotation domain lattice.

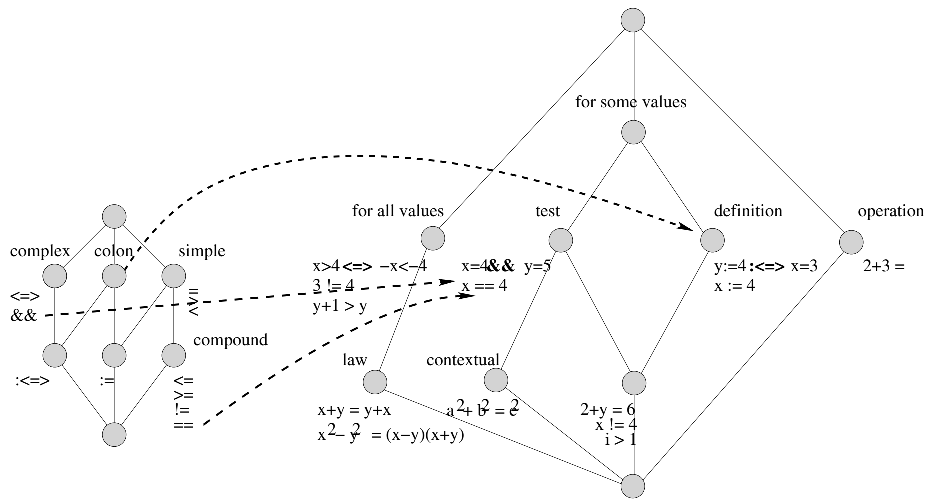


Fig. 3. Equation, assignment and comparison operators

In Fig. 4 the denotation domain lattice has been restructured in order to support more interpretation instances which do not point towards the top concept. It should be noted that the lattice on the right-hand side is shown as an incomplete ‘nested line diagram’ which has nodes missing. The restructuring of the lattice has also used ideas from APOS Theory (Dubinsky & Mcdonald, 2002) according to which learning of mathematical concepts often progresses from action- to progress- to object-level understanding. At an action level understanding students perform operations without really knowing much about them. Prediger’s (2010) operational use of the equals sign appears to be action level. With respect to the representamen domain lattice it appears reasonable to separate the interpretation images of complex and simple representamens. From an APOS viewpoint this could correspond to a difference between process and object level understanding. At an object level, a concept becomes itself reified and part of another concept. In the examples with more than one operator, the simple operators become objectified. For example in $x = 4 \,\&\&\, y = 5$, the Boolean operator && is primary whereas its left and right operands are only evaluated with respect to their truth values.

Altogether the denotation domain lattice in Fig. 4 might be an example of what is called a ‘genetic decomposition’ in APOS Theory. A student’s conceptual learning should move from the top of the lattice to the bottom. A complete understanding of the comparison operators is achieved when a student knows that the operators are used with different meanings and knows exactly what each operator means in the context it is represented in. Depending on what a student says about one of the operators and in particular depending on what kinds of errors a student makes when using an operator, the student’s conceptual stage at that point could be pinpointed in the denotation domain lattice.

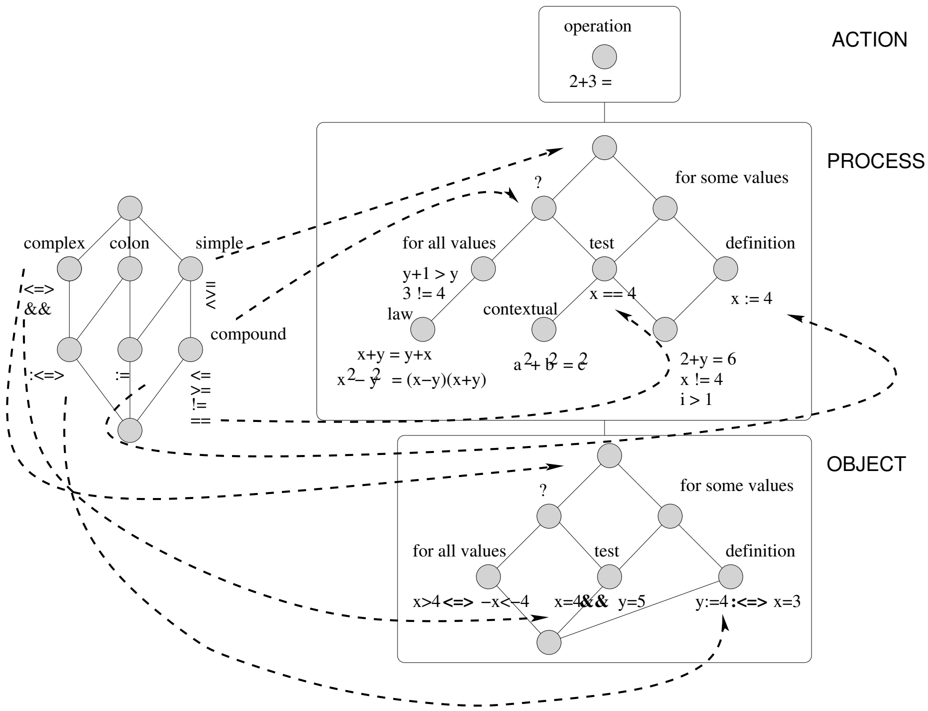


Fig. 4. Remodelling of the denotation domain lattice from Fig. 3

7 Conclusion

This paper shows how the modelling of some of the concepts of a domain in combination and in contrast with a modelling of the representamens that are used for the domain can serve as a tool for analysing the knowledge state a student is in. If a student uses terms incorrectly then the student's interpretation must be different from the teacher's interpretation. From a theoretical viewpoint, mapping the signs uttered by a student into a denotation domain lattice shows possible misconceptions on the student's part. From a practical viewpoint, this implies that it is important to observe how a student talks about denotations while he or she performs tasks within the domain. For example, a student could be asked to perform a mathematical calculation and discuss it at the same time. In that manner any misalignment between representamens and denotations might become apparent for the teacher. If a student just writes a textual exam or just performs a calculation, it could be possible that the student just reproduces material he or she has memorised without actually understanding it.

The examples in the previous sections show methods that use denotation domain lattices for representing prerequisite conditions within domain knowledge similar to Knowledge Space Theory. Furthermore, a denotation domain lattice is shown that could serve as a genetic decomposition in the sense of APOS Theory which is an established constructivist theory of mathematics education. Thus this paper enlarges the area of possible applications of SCA from analysing programming code (as presented by Priss (2015)) to teaching and learning of formal representations. In the future, we plan to explore further applications of SCA for educational tasks but also with respect to structural analysis of formal representations, for example, with respect to object-oriented programming and XML.

References

1. Albert, D.; Lukas, J. (eds.) (1999). *Knowledge Spaces: Theories, Empirical Research, Applications.* Mahwah, NJ: Lawrence Erlbaum Associates.
2. Dubinsky, Ed.; A. Mcdonald, Michael (2002). *APOS: A Constructivist Theory of Learning in Undergraduate Mathematics Education Research.* ICMI Study Series Volume 7, p. 275-282.
3. Falmagne, J.-C., Albert D., Doble, D., Eppstein D., Hu X. (2013). *Knowledge Spaces: Applications in Education.* Berlin: Springer-Verlag.
4. Ganter, Bernhard; Wille, Rudolf (1999). *Formal Concept Analysis. Mathematical Foundations.* Berlin-Heidelberg-New York: Springer.
5. Goguen, Joseph (1999). *An introduction to algebraic semiotics, with application to user interface design.* Computation for metaphors, analogy, and agents. Springer Berlin Heidelberg, p. 242-291.
6. Prediger, Susanne (2010). *How to develop mathematics-for-teaching and for understanding: the case of meanings of the equal sign.* J. Math. Teacher Educ., 13, p. 73-93.
7. Priss, Uta (1998). *Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases.* (PhD Thesis) Verlag Shaker, Aachen.
8. Priss, Uta (2004). *Signs and Formal Concepts.* In: Eklund (ed.), Concept Lattices: Second International Conference on Formal Concept Analysis, Springer Verlag, LNCS 2961, 2004, p. 28-38.
9. Priss, Uta; Riegler, Peter; Jensen, Nils (2012). *Using FCA for Modelling Conceptual Difficulties in Learning Processes.* In: Domenach; Ignatov; Poelmans (eds.), Contributions to the 10th International Conference on Formal Concept Analysis (ICFCA 2012), p. 161-173.
10. Priss, Uta (2015). *An Introduction to Semiotic-Conceptual Analysis with Formal Concept Analysis.* In: Yahia; Konecny (eds.), Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, 2015, p. 135-146.