Combining Two Formalism for Reasoning about Concepts*

(extended abstract)

N.V. Shilov, N.O. Garanina, and I.S. Anureev

A.P. Ershov Institute of Informatics Systems,
Lavren'ev av., 6, Novosibirsk 630090, Russia,
{shilov, garanina, anureev}@iis.nsk.su

Abstract. There are two major formalisms that are developed around concepts. The first one is Formal Concept Analysis (FCA) by R. Wille and B. Ganter. Roughly speaking, FCA is an extension of algebraic Lattice Theory for knowledge representation. The second formalism, Description Logic (DL), goes back to the universal terminological logic by P.F. Patel-Schneider. It is closely related to modal and program logics. DL is widely used for ontology research, design, and implementation. Since both formalism use concepts and are used for closely related purposes, it is very natural to compare and combine them.
In this paper we introduce and study variants of DL extended by three constructs motivated by FCA. Intentional semantics of two of the new constructs are new modalities that correspond to 'intent' and 'extent' (two major algebraic constructions of FCA). The third new construct is a connective that is designed to express the 'formal concept' property. If $\mathcal{L}$ is a variant of DL then we call $\mathcal{L}$ extended by these new constructs by $\mathcal{L}$ for FCA and denote this logic by $\mathcal{L}$/FCA.
We compare expressive powers of $\mathcal{L}$/FCA and $\mathcal{L}(\neg, -)$ – another variant of $\mathcal{L}$ extended by role complement $\neg$ and inverse $-$ simultaneously. We demonstrate that $\mathcal{L}$/FCA can be expressed in $\mathcal{L}(\neg, -)$. It implies that for the basic description logic $\mathcal{ALC}$, $\mathcal{ALC}$/FCA is decidable.

1 Basic Description Logics

Description logics [2] has originated from the universal terminological logic [9]. There exists many variants of description logics, but we will define only some of them in this section[1].

Definition 1. *Syntax of every description logic is constructed from disjoint alphabets of concept, role, and object symbols CS, RS, and OS, respectively. The*

* This research is supported in parts by joint grant RFBR 05-01-04003-a - DFG project COMO, GZ: 436 RUS 113/829/0-1, by grant RFBR 06-01-00464-a, and Integration Grant n.14 Siberia Branch, Russian Academy of Science.

1 We give detailed definition of description logics for avoiding ambiguities, since there exists some difference in syntax notation between different research groups.

sets of concept terms (or concepts) CT and role terms (or roles) RT are defined by induction. The usual definition admits the following clauses[2].

- *(Concept terms)*
  - *the top concept* $\top$ *and the bottom concept* $\bot$ *are concept terms;*
  - *any concept symbol is a concept term;*
  - *for any concepts* X *and* Y *their union* $(X \sqcup Y)$ *and intersection* $(X \sqcap Y)$ *are concept terms;*
  - *for any concept* X *its complement* $(\neg X)$ *is a concept term;*
  - *for any role* R *and any concept* X *the universal* $(\forall R.\ X)$ *and existential* $(\exists R.\ X)$ *restrictions are concept terms;*
- *(Role terms)*
  - *the top role* ∇ *and the bottom role* $\triangle$ *are role terms;*
  - *any role symbol is a role term;*
  - *for any roles* R *and* S *their union* $(R \sqcup S)$*, intersection* $(R \sqcap S)$*, and composition* $(R \circ S)$ *are terms;*
  - *for any role* R *its complement* $(\neg R)$*, inverse* (R^-)*, and transitive closure* (R^+) *are role terms.*

Concept and role terms altogether form the set of terminological expressions.

Definition 2.

- *For any concepts* X *and* Y*, any roles* R *and* S *the following expressions are called terminological sentences:* $X \dot{\sqsubseteq} Y$, $X \doteq Y$, $R \dot{\sqsubseteq} S$*, and* $R \doteq S$*. A TBox is a set of terminological sentences.*
- *For any concept* X*, any role* R*, and any object symbols* a *and* b *the following expressions are called assertional sentences: a concept assertion* $a : X$ *and a role assertion* $(a, b) : R$*. An ABox is a set of assertional sentences.*
- *A knowledge base is a finite set of terminological and assertional sentences. Every knowledge base consists of an appropriate TBox and ABox.*

Definition 3. *Semantics of any description logic is defined in Kripke-like terminological interpretations. Every terminological interpretation is a pair* (D, I)*, where* D *is a set (that is called domain) and* I *is a mapping (that is called interpretation function). This function maps object symbols to elements of* D*, concept symbols to subsets of* D*, role symbols to binary relations on* D*:* $I = I_{OS} \cup I_{CS} \cup I_{RS}$*, where* $I_{OS} : OS \to D$*,* $I_{CS} : CS \to 2^D$*, and* $I_{RS} : RS \to 2^{D\times D}$*. The unique name assumption holds for this function:* $I(a) \neq I(b)$ *for all different object symbols* a *and* b*. The interpretation function can be extended to all terminological expressions as follows.*

- *(Concept semantics)*
  - $I(\top) = D$ *and* $I(\bot) = \emptyset$*;*
  - $I(X \sqcup Y)) = I(X) \cup I(Y)$ *and* $I(X \sqcap Y) = I(X) \cap I(Y)$*;*
  - $I(\neg X) = D \setminus I(X)$*;*

[2] We omit some constructs that can be treated as derived features.

- $I(\forall R.\ X) = \{s \in D : \forall t \in D(\textit{if } (s,t) \in I(R) \textit{ then } t \in I(X))\}$,
  $I(\exists R.\ X) = \{s \in D : \exists t \in D((s,t) \in I(R) \textit{ and } t \in I(X))\}$;

– *(Role semantics)*
  - $I(\nabla) = D^2$ *and* $I(\triangle) = \emptyset$;
  - $I(R \sqcup S) = I(R) \cup I(S)$, $I(R \sqcap S) = I(R) \cap I(S)$, $I(R \circ S) = I(R) \circ I(S)$ *(righthand side '$\circ$' is composition of binary relations);*
  - $I(\neg R) = D^2 \setminus I(R)$, $I(R^-) = (I(R))^-$, *and* $I(R^+) = (I(R))^+$ *(righthand side '$^+$' is transitive closure of binary relations).*

Definition 4. *Semantics of sentences is defined in terminological interpretations in terms of satisfiability relation as follows:*

– $(D, I) \models a : X$ *iff* $I(a) \in I(X)$;
– $(D, I) \models X \dot\sqsubseteq Y$ *iff* $I(X) \subseteq I(Y)$;
– $(D, I) \models X \doteq Y$ *iff* $I(X) = I(Y)$;
– $(D, I) \models (a, b) : R$ *iff* $(I(a), I(b)) \in I(R)$;
– $(D, I) \models R \dot\sqsubseteq S$ *iff* $I(R) \subseteq I(S)$;
– $(D, I) \models R \doteq S$ *iff* $I(R) = I(S)$.

This satisfiability relation can be extended on knowledge bases in a natural way: for any knowledge base $KBase$, $(D, I) \models KBase$ *iff* $(D, I) \models \phi$ *for every sentence* $\phi \in KBase$. *In the case of* $(D, I) \models KBase$, *the terminological interpretation* (D, I) *is said to be a (terminological) model for the knowledge base* $KBase$. *Let us say that a knowledge base* $KBase$ *entails a sentence* ψ *(and write*[3] $Kbase \models \psi$*) iff* $(D, I) \models \psi$ *for every model* (D, I) *for* $KBase$.

Definition 5. *A concept* X *is said to be coherent (or satisfiable) with respect to a knowledge base* $KBase$ *iff there exists a terminological model* (D, I) *for* $KBase$ *such that* $I(X)$ *is not empty. A knowledge base* $KBase$ *is said to be satisfiable iff the top concept* $\top$ *is coherent with respect to* $KBase$.

Satisfiability problem is to check for input knowledge base $KBase$ whether it is satisfiable or not. It is well known that the problem is undecidable [2] for description logic that admits all syntax constructs that are enumerated in the definition 1. This undecidability boundary drives many researchers to study of description logics with decidable satisfiability problem. An important role in these studies belongs to a fragment that is called Attribute Language with Complements ($\mathcal{ALC}$) [11]. In simple words, $\mathcal{ALC}$ adopts role symbols as the only role terms, concept symbols – as elementary concept terms, and permits 'boolean' constructs '$\neg$', '$\sqcup$', '$\sqcap$', universal and existential (but non-limited) restrictions '$\forall$' and '$\exists$' as the only concept constructs. The formal definition follows.

Definition 6. $\mathcal{ALC}$ *is a fragment of DL that comprises concepts that are defined by the following context-free grammar:*

$$C_{\mathcal{ALC}} ::= \\ CS|\top|\bot|(\neg C_{\mathcal{ALC}})|(C_{\mathcal{ALC}} \sqcup C_{\mathcal{ALC}})|(C_{\mathcal{ALC}} \sqcap C_{\mathcal{ALC}})|(\forall RS.\ C_{\mathcal{ALC}})|(\exists RS.\ C_{\mathcal{ALC}})$$

[3] Let us write '$\models \psi$' instead of '$\emptyset \models \psi$' when knowledge base is empty.

where metavariables CS and RS represent any concept and role symbols, respectively. Semantics of $\mathcal{ALC}$ is defined in the standard way in accordance with Definition 3.

Many description logics can be defined as extensions of $\mathcal{ALC}$ by concept and/or role constructs. For example, the website [13] uses the following approach: for any collection of concept and/or role constructs $C\&R$, let $\mathcal{ALC}(C\&R)$ be a 'closure' of $\mathcal{ALC}$ that admits all concept and/or role constructs in $C\&R$. Formal definitions follows.

Definition 7. *A variant of DL is a description logic $\mathcal{L}$ with syntax that*

- *contains all concept and role symbols CS and RS,*
- *is closed under concept constructs '$\neg$', '$\sqcup$', '$\sqcap$', '$\forall$' and '$\exists$'.*

From the viewpoint of the above definition, $\mathcal{ALC}$ is the smallest variant[4] of DL.

Definition 8. *Let $\mathcal{L}$ be a variant of DL and $C\&R$ be a collection of concept and/or role constructs. Then let $\mathcal{L}(C\&R)$ be the smallest variant of DL that includes $\mathcal{L}$ and is closed under all constructs in $C\&R$.*

For instance, $\mathcal{ALC}(\neg, -)$ is an extension of $\mathcal{ALC}$ where any role symbol can be negated and/or inverted. This variant of DL has decidable satisfiability problem [11, 7].

2 Integrating FCA operations to DL

Basic Formal Concept Analysis (FCA) definitions below follow monograph [3].

Definition 9. *A formal context is a triple (O, A, B) where O and A are sets of 'objects' and 'attributes' respectively, and $B \subseteq O \times A$ is a binary relation connecting objects and attributes. Let us say that a formal context (O, A, B) is homogeneous[5] iff $O = A$, i.e. the set of objects coincide with the set of attributes.*

For example, for every terminological interpretation (D, I) and every role r one can define a formal context $(D, D, I(r))$. It implies that every terminological interpretation (D, I) defines a family of homogeneous formal contexts $(D, D, I(R))$ indexed by role symbols $R \in RS$ or by role terms $R \in RT$.

Vise verse, there is a number of ways how to define a terminological interpretation for given formal contexts. For example, if we have a family of formal contexts (O_j, A_j, B_j) indexed by elements of some set J, then we can adopt the set of indices J as the alphabet role symbols RS, a set of symbols $\{o_j, a_j : j \in J\}$ as the alphabet of concept symbols CS, and define a terminological interpretation (D, I) where

[4] Of course, 'smaller' description logics can be defined and examined by means of stronger syntax restrictions.

[5] 'Homogeneous' is our own non-standard FCA term.

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
01: Fun95							x																		
02: God93																									
03: God95		x																							
04: God98		x	x				x																		
05: Huc99		x																							
06: Huc02																x			x						
07: Kro94																									
08: Kui00											x		x	x	x	x		x							x
09: Leb99		x																							
10: Lin95							x																		
11: Lin97	x						x							x											
12: Sah97			x								x		x												
13: Sif97											x	x		x											
14: Sne96							x			x															
15: Sne98C	x						x			x	x		x	x											
16: Sne98R		x					x				x		x	x	x										
17: Sne99	x	x		x			x				x		x	x	x	x									
18: Sne00S							x				x		x	x		x	x								x
19: Sne00U		x		x			x				x			x	x	x									
20: Str99														x	x	x									
21: Til03S											x					x									
22: Til03T																									
23: Ton99							x				x		x	x											
24: Tone01							x				x		x	x			x								
25: Van98											x		x		x	x									

Table 1. Context *Citations*

- $D = \cup_{j\in J}(O_j \cup A_j)$,
- $I(j) = B_j \subseteq (O_j \times A_j) \subseteq D \times D$ for every $j \in J$,
- $I(a_j) = A_j \subseteq D$ and $I(o_j) = O_j \subseteq D$ for every $j \in J$.

Table 1 represents an example of a homogeneous formal context *Citations* from [10]. It represents citations between papers in some collection. In this particular case the set of objects (rows) and the set of attributes (columns) are both equal to [1..25].

Two basic algebraic operations for formal contexts are upper and lower derivations. These operations are used in the definition of a notion of a formal concept, its extent and intent.

Definition 10. *Let (O, A, B) be a formal context. For every set of objects $X \subseteq O$ its upper derivation $X^{\uparrow}$ is the following set of attributes*

$$\{t \in A : \text{ for every } s \in O, \text{ if } s \in X \text{ then } (s,t) \in B\},$$

i.e. the collection of all attributes that are satisfied by all objects in X simultaneously. For every set of attributes $Y \subseteq A$ its lower derivation $Y^{\downarrow}$ is the following set of objects

$$\{s \in O : \text{ for every } t \in A, \text{ if } t \in Y \text{ then } (s,t) \in B\},$$

i.e. the collection of all objects that satisfy to all objects in Y simultaneously. A formal concept is a pair (Ex, In) such that $Ex \subseteq O$, $In \subseteq A$, and $Ex^{\uparrow} = In$, $In^{\downarrow} = Ex$; components Ex and In of the formal concept (Ex, In) are called its extent and its intent respectively.

For example, $\{3,4\}^{\uparrow} = \{2\}$ and $\{2\}^{\downarrow} = \{3,4,5,9,16,17,19\}$ in the context *Citations*. Pair $(\{1,4,10,11,14,15,16,17,18,19,23,24\}\ ,\ \{7\})$ is an example of a formal concept in the formal context *Citations*.

Definition 11. *Let $\mathcal{L}$ be a variant of DL. Then let $\mathcal{L}/FCA$ be a variant of DL that is the closure of $\mathcal{L}$ with respect to two new formula constructors for the upper and lower derivatives. Syntax of these two constructs is as follows: for every role term R and every concept term X let $(X^{\uparrow R})$ and $(X^{\downarrow R})$ be concept terms too. They are read as 'upper derivative of X with respect to R' and, respectively, as 'lower derivative of X with respect to R'. For every terminological interpretation (D,I),*

- $I(X^{\uparrow R}) = \{t :$ *for every* $s \in D$, *if* $s \in I(X)$ *then* $(s,t) \in I(R)\}$, *i.e. the upper derivation of $I(X)$ in a homogenous formal context $(D,D,I(R))$;*
- $I(X^{\downarrow R}) = \{s :$ *for every* $t \in D$, *if* $t \in I(X)$ *then* $(s,t) \in I(R)\}$, *i.e. the lower derivation of $I(X)$ in a homogenous formal context $(D,D,I(R))$.*

In particular, $\mathcal{ALC}/FCA$ is an extension of $\mathcal{ALC}$ where both derivative constructors are allowed.

Proposition 1.

1. *Let $\mathcal{L}$ be a variant of DL. For every role and concept terms within $\mathcal{L}$ the following concepts of $\mathcal{L}/FCA$ and $\mathcal{L}(\neg,-)$ are equivalent (i.e. have equal semantics in every terminological interpretation):*
   (a) $X^{\uparrow R}$ and $\forall\neg R^{-}.\ \neg X$,
   (b) $X^{\downarrow R}$ and $\forall\neg R.\ \neg X$.
2. *$\mathcal{L}/FCA$ can be expressed in $\mathcal{L}(\neg,-)$ with linear complexity, i.e. every concept X in $\mathcal{L}/FCA$ is equivalent to some concept Y in $\mathcal{L}(\neg,-)$ that can be constructed in linear time.*

Proposition 2. *For every terminological interpretation (D,I), every concept terms X and Y of DL, and every role term R of DL the following holds:*

a pair of sets $(I(X)\ ,\ I(Y))$ is a formal concept
in the homogeneous formal context $(D,D,I(R))$
$$\Updownarrow$$
(D,I) is a terminological model
for the following two terminological sentences
$$X^{\uparrow R} \doteq Y \text{ and } Y^{\downarrow R} \doteq X.$$

Proposition 2 makes sense to the following definition.

Definition 12.
For any concept terms X and Y, for any role term R, let $(X,Y)FC(R)$ be a 'terminological sentence' that is a shorthand (a notation or an abbreviation) for the following pair of standard terminological sentences $X^{\uparrow R} \doteq Y$ and $Y^{\downarrow R} \doteq X$. This notation is read as '(X,Y) is a Formal Concept with respect to R'.

The above propositions and the decidability of the satisfiability problem for $\mathcal{ALC}(\neg,-)$ [7] together imply the next corollary.

Corollary 1. *The satisfiability problem for $\mathcal{ALC}/FCA$ (including terminological sentences for formal concepts) is decidable.*

3 Concluding Remarks

The primary target of our research was to make explicit relations between two formalisms for reasoning about concepts. The first formalism, Formal Concept Analysis (FCA), is of algebraic nature. The second one, Description Logic (DL), is of logical nature. We have demonstrated in the present paper that FCA can be 'absorbed' by DL at least from viewpoint of 'abstract' expressive power. It implies that any collection set-theoretic (in)equalities written in terms of uninterpreted symbols for individual objects and attributes, for sets of objects and attributes, for formal contexts and concepts, with aid of set-theoretic operations, FCA operations for upper and lower derivative, intent and extent operations, can be easy translated to a description logic knowledge base, so that the base is satisfiable iff there is a formal context where all these (in)equalities realize simultaneously. Since the satisfiability problem is decidable for many Description Logics, the realization problem for collections of (in)equalities of this kind can be done (as a rule) automatically (i.e. by some algorithm).

At the same time in the present paper we give a partial answer to a question from [12], whether a variant of Propositional Dynamic Logic (PDL) extended by upper and lower derivations for atomic programs is decidable. PDL [5, 6] has been introduced by M.J. Fischer and R.E. Ladner as an extension of the classical propositional logic and propositional modal logic K for reasoning about partial correctness of structured nondeterministic programs. Many variants of PDL have been studied extensively especially from the viewpoint of decidability and axiomatizability. In particular, recently C. Lutz and D. Walther [8] have proved that PDL with complement of atomic programs is decidable in exponential time (while it is well known that in general case PDL with complement is undecidable).

Paper [12] has introduced and studied PDL/FCA – a variant of PDL extended by modalities inspired by Formal Concept Analyses (FCA). Formal semantics of these modalities is upper and, respectively, lower derivations. (Please refer [12] for discussion on utility of these modalities for program specification and verification.)

Paper [12] has proved that PDL/FCA is more expressive than PDL, and has interpreted a fragment of PDL/FCA without upper derivation in PDL with complement. It implies decidability of PDL extended by extent of atomic programs with exponential upper bound (since PDL with complement of atomic programs is decidable [8]). It remains an open question whether PDL/FCA (without any restriction for upper and lower derivations) is decidable and what is the expressive power of this logic (with respect to PDL with complement). But now (due to Corollary 1) we can claim that a fragment of PDL/FCA with atomic programs is decidable (since this fragment is equal to $\mathcal{ALC}$/FCA).

There is a number of research papers on combination of Formal Concept Analysis with Description Logic for better knowledge processing[6]. But there are

[6] For instance please refer a recent paper [1], but a survey of this topic is out of scope of the present paper.

few papers on comparison and integration of both formalism in one. We can point just a single one [10] related to this topic. The cited paper has attempted to develop in the framework of FCA an algebraic operation inspired by the universal restriction of DL and to demonstrate an utility of it for analysis of relational data. In accordance with [10], the attempt has resulted in a so-called 'Relational Concept Analysis' that had been implemented in an open platform Galicia for lattices.

Acknowledgement: We would like to thank Prof. Karl Erich Wolff for fruitful discussions of the research and for the comments on the draft of this paper. We also would like to thanks an anonymous referee for pointing a paper [4], where a construct corresponding to the lower derivation operator has been considered before our research in a framework of description logics.

References

1. Baader F., Ganter B., Sattler U., and Sertkaya B. *Completing Description Logic Knowledge Bases using Formal Concept Analysis.* In Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07). AAAI Press, 2007.
2. Baader F., Calvanese D., Nardi D., McGuinness D., and Patel-Schneider P., editors. *The Description Logic Handbook: Theory,Implementation and Applications.* Cambridge University Press, 2003.
3. Ganter B., Wille R. *Formal Concept Analysis.* Mathematical Foundations. Springer Verlag, 1996.
4. Givan R., McAllester D., and Shalaby S. *Natural language based inference procedures applied to schubert's steamroller.* In AAAI-91, p. 915–920. Morgan Kaufmann Publishers.
5. Fischer M.J. and Ladner R.E. Propositional dynamic logic of regular programs. J. Comput. Syst. Sci., 18(2):194-211, 1979.
6. Harel D., Kozen D., Tiuryn J. Dynamic Logic (Foundations of Computing). MIT Press, 2000.
7. Hustadt U, Schmidt R.A., and Georgieva L. *A Survey of decidable first-order fragments and description logics.* J. Relational Methods in Computer Science, Vol. 1, 2004, pp. 251-276.
8. Lutz C. and Walther D. PDL with Negation of Atomic Programs. Journal of Applied Non-Classical Logic, 15(2):189-214, 2005.
9. Patel-Schneider P. F. *Decidable, Logic-Based Knowledge Representation.* PhD thesis, Univ. Toronto, 1987.
10. Rouane A.H., Huchard M., Napoli A., Valtchev P. *A proposal for combining formal concept analysis and description logics for mining relational data.* Lecture Notes in Artificial Intelligence, Vol. 4390, 2007, pp. 51-65.
11. Schmidt-Schauß M., and Smolka G. *Attributive concept descriptions with complements.* J. Artificial Intelligence, Vol. 48, 1991, pp. 1-26.
12. Shilov N.V., Garanina N.O., Anureev I.S. Combining *Propositional Dynamic Logic with Formal Concept Analysis*. Concurrency, Specification and Programming CS&P'2006. Volume 2: Specification. Humboldt-Universitat zu Berlin, Informatik-Bericht Nr.206, 2006, p.152-161.
13. Zolin E. Complexity of reasoning in Description Logics. Web-resource available at URL `http://www.cs.man.ac.uk/~ezolin/logic/complexity.html`.