A Semantic Algebra for Modularized Description Logics Knowledge Bases

Krzysztof Goczyła, Aleksander Waloszek, Wojciech Waloszek

Gdańsk University of Technology, Department of Software Engineering,
ul. Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland
{kris,alwal,wowal}@eti.pg.gda.pl

1 Introduction

Subsequent years bring increasing attention into the area of modularization of Description Logics (DL) knowledge bases. The main motivation is the hope to reach the maturity of the collaborative ontology development and reuse process comparable to the one achieved by software engineering methods in the case of software modules. As a consequence many methods and techniques of ontology decomposition and merging have been devised.

From the mathematical point of view a very useful method to describe a system of complex operations on ontologies would be an algebra. There were many attempts to create algebras for ontology modules. Two notable examples come from [1] and [8]. These algebras treat modules as sets of sentences (as in the case of [1]) or more complex syntactic structures (like in the case of [8]). We found this approach unsatisfying, and decided to turn toward algebras of interpretations. This decision was motivated by two important observations. The first observation is that the most straightforward definitions of syntactic algebras do not seem to capture the intended meaning, which is to some extent a consequence of the fact that the different sets of sentences may have the same interpretations (we further discuss this issue in Sec. 4).

The second observation is that semantic approach to defining algebra can be regarded more fundamental than syntactic (or deductive) ones. By using sentences we are in fact eliminating unwanted interpretations. Therefore, by operating on sets of sentences, we in fact operate on interpretations: every "syntactic" algebra determines a "semantic" algebra. Study of an algebra operating on interpretations, may be therefore treated as a good theoretical background for design of any practical "syntactic" algebra operating on linguistic representations.

In this paper we propose an algebra operating on semantic modules. We called our formalism *s-module* algebra ("s" stands for "semantic"). Section 3.1 contains definitions of s-module and algebraic operations. In Section 3.2 we show that our algebra is fully homomorphic to relational and cylindric algebras, which gives us a very strong framework for exploring its properties. In Section 4 we discuss the role of s-module algebra and relevance of choice of operators. Section 5 concludes the paper.

2 Preliminaries

In the further discussion we assume that we use an arbitrary chosen description logics $\mathcal{L}$.

We exploit the notion of a signature, denoted $\mathbf{S} = \mathbf{AC} \uplus \mathbf{AR} \uplus \mathbf{Ind}$ and assume that for any $\mathcal{L}$ it is a disjoint union of concept names ($\mathbf{AC}$), role names ($\mathbf{AR}$), and individual names ($\mathbf{Ind}$). To refer to specific part of the signature $\mathbf{S}$ we use the notation $\mathbf{AC}(\mathbf{S})$, $\mathbf{AR}(\mathbf{S})$, $\mathbf{Ind}(\mathbf{S})$ respectively. We assume that there exist sets of acceptable concepts, roles, and individual names, which we denote $\mathcal{N}_{\mathrm{AC}}$, $\mathcal{N}_{\mathrm{AR}}$, $\mathcal{N}_{\mathrm{Ind}}$ respectively ($\mathbf{AC} \subseteq \mathcal{N}_{\mathrm{AC}}, \mathbf{AR} \subseteq \mathcal{N}_{\mathrm{AR}}, \mathbf{Ind} \subseteq \mathcal{N}_{\mathrm{Ind}}$). The full signature $\widehat{\mathbf{S}}$ contains all the names ($\widehat{\mathbf{S}} = \mathcal{N}_{\mathrm{AC}} \uplus \mathcal{N}_{\mathrm{RA}} \uplus \mathcal{N}_{\mathrm{Ind}}$).

Chosen description logics $\mathcal{L}$ determines the set of (possibly complex) concepts, roles and individuals that can be built using the operators of $\mathcal{L}$ and names from a signature $\mathbf{S}$. We denote these sets with $\mathcal{L}_{\mathrm{C}}(\mathbf{S})$, $\mathcal{L}_{\mathrm{R}}(\mathbf{S})$, $\mathcal{L}_{\mathrm{I}}(\mathbf{S})$ respectively. For instance, if $\mathcal{L}$ is $\mathcal{ALC}$, $\mathcal{L}_{\mathrm{C}}(\mathbf{S}) ::= A \,|\, \neg C \,|\, C \sqcap D \,|\, C \sqcup D \,|\, \exists R.C \,|\, \forall R.C \,|\, \top \,|\, \bot$ where $A \in \mathbf{AC}(\mathbf{S})$, $C, D \in \mathcal{L}_{\mathrm{C}}(\mathbf{S})$, $R \in \mathcal{L}_{\mathrm{R}}(\mathbf{S})$.

An (base) interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a set called the domain of the interpretation and $\cdot^{\mathcal{I}}$ is an interpretation function assigning each $A \in \mathcal{N}_{\mathrm{AC}}$ a set $A^{\mathcal{I}} \subseteq \Delta$, each $R \in \mathcal{N}_{\mathrm{AR}}$ a binary relation $R^{\mathcal{I}} \subseteq \Delta \times \Delta$, and each $a \in \mathcal{N}_{\mathrm{Ind}}$ an element $a^{\mathcal{I}}$ of the domain $\Delta^{\mathcal{I}}$. Note that following [4] we assume that every interpretation interprets all the base symbols. Each description logic $\mathcal{L}$ establishes its own rules of extending the base interpretation to complex concepts (e.g. in $\mathcal{ALC}$ $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$), however in this paper we primarily focus on base interpretations (this is not really a restriction, since for every description logic there is one-to-one correspondence between base and extended interpretation). The class of all (base) interpretations we denote as $\mathfrak{I}$.

Each description logic allows for formulating axioms and assertions (sentences), and define the conditions under which a specific interpretation $\mathcal{I}$ satisfies a particular sentence. We denote the set of axioms and assertions which can be built from the given signature $\mathbf{S}$ as $\mathcal{L}(\mathbf{S})$. The fact that the interpretation $\mathcal{I}$ satisfies a given sentence $\alpha \in \mathcal{L}(\mathbf{S})$ (is its model) is denoted as $\mathcal{I} \models \alpha$.

From the point of view of s-module algebra some selected operations on interpretations are particularly interesting. These operations are: interpretation projection (to the given signature) and interpretation restriction (to the given domain). Given the set of interpretations $\mathbf{I}$ and the signature $\mathbf{S}$ we define the *interpretation projection* $\mathbf{I}|\mathbf{S}$ as $\left\{\mathcal{J} \in \mathfrak{I} : \exists \mathcal{I} \in \mathbf{I} : \Delta^{\mathcal{I}} = \Delta^{\mathcal{J}} \wedge \forall X \in \mathbf{S} : X^{\mathcal{I}} = X^{\mathcal{J}}\right\}$. We simplify the notation and instead of $\{\mathcal{I}\}|\mathbf{S}$ we write $\mathcal{I}|\mathbf{S}$ remembering that we will obtain a set of interpretations as the result. Note that always $\mathbf{I} \subseteq \mathbf{I}|\mathbf{S}$.

The notion of signature restriction is similar to coincidence of two interpretations on signature $\mathbf{S}$ [4]. We say that $\mathcal{I}$ and $\mathcal{J}$ *coincide on a signature* $\mathbf{S}$ (notation $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$) if $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and $X^{\mathcal{I}} = X^{\mathcal{J}}$ for every $X \in \mathbf{S}$. Note that $\mathbf{I}|\mathbf{S} = \{\mathcal{J} \in \mathfrak{I} : \exists \mathcal{I} \in \mathbf{I} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}\}$.

The *restriction* of an interpretation $\mathcal{I}$ to the domain Δ', denoted by $\mathcal{I} \cap \Delta'$, is an interpretation $\mathcal{I}' = (\Delta', \cdot^{\mathcal{I}'})$ such that $X^{\mathcal{I}'} = X^{\mathcal{I}} \cap \Delta'$ for every X.

3 S-module algebra

3.1 Basic definitions

The basic element of the universe of the s-module algebra is a semantic module defined below.

Definition 1 (s-module). A s-module $M = (\mathbf{S}, \mathbf{W})$ *is a pair of a signature* $\mathbf{S}$ *(called a signature of the s-module) and a class* $\mathbf{W}$ *of interpretations, such that* $\mathbf{W}|\mathbf{S} = \mathbf{W}$*. The two parts of* M *are denoted respectively as* $\mathbf{S}(M)$ *and* $\mathbf{W}(M)$*. Each interpretation from* $\mathbf{W}$ *is called a model of* M*.*

According to this definition each module simply consists of all its models. Models of a specific s-module M cannot express relationships between concepts, roles, and individuals from outside the signature $\mathbf{S}$, which is guaranteed by the condition $\mathbf{W}|\mathbf{S} = \mathbf{W}$. We say that s-module satisfies a particular sentence $\alpha \in \mathcal{L}$, which we denote $M \models \alpha$, iff $\forall \mathcal{I} \in \mathbf{W}(M) : \mathcal{I} \models \alpha$.

Models of even very simple s-modules form a proper class because of the fact that any domain set can be used in each interpretation from $\mathbf{W}$. In practice we may, without much loss of generality of the formalism, assume that all the domains of the interpretations in $\mathbf{W}$ are subsets of some given set $\mathbf{\Delta}$. Under this assumption $\mathbf{W}$ is a set. Note that this assumption differs slightly from fixed-domain assumption as we do not fix the domain but restrict it to be a subset of the given (possibly very large) set.

From the point of view of a user of the algebra, each model may be perceived as a one "possible world" or (to avoid confusion with modal logics) just a "possibility". The s-module algebra provides the user with operators for manipulating sets of "possibilities".

In the process of combining knowledge from different s-modules we distinguished three categories of activities:

1. Operations on signatures of the modules, e.g. extending the vocabulary,
2. Operations relating on the whole classes of models of s-modules without specifying new relationships between concepts, roles, and individuals (terms),
3. Operations introducing new relationships between terms.

For each category we introduced a set of operators. Below we give their formal definitions. For the sake of simplicity, we assume that description logic $\mathcal{L}$ and domain set $\mathbf{\Delta}$ are chosen (in fact none of these assumptions is indispensable). In all definitions we denote the set of all modules as $\mathfrak{M}$, the set of all signatures as $\mathfrak{S}$, and the set of all interpretations as $\mathfrak{I}$.

To support the first category of activities we introduce the operations of signature *extension* (ϵ), *projection* (π) and *rename* (ρ), which allow for adding, removing and changing names in the vocabulary.

$$\begin{array}{ll}
\epsilon_{\mathbf{S}}(M) = (\mathbf{S}(M) \cup \mathbf{S}, \mathbf{W}(M)) & \mathbf{S} \in \mathfrak{S} \\
\pi_{\mathbf{S}}(M) = (\mathbf{S}, \mathbf{W}(M)|\mathbf{S}) & \mathbf{S} \subseteq \mathbf{S}(M) \\
\rho_{\gamma}(M) = (\gamma(\mathbf{S}(M)), \gamma(\mathbf{W}(M))) & \gamma \text{ is a signature mapping}
\end{array}$$

Extension operation extends a signature of a given module M by names in a given signature $\mathbf{S}$. The allowed set of interpretations is preserved, and so are the relationships between original concepts, roles, and individuals (e.g. if $M \models \alpha$, $\alpha \in \mathcal{L}(\mathbf{S}(M))$, then $\epsilon_{\mathbf{S}}(M) \models \alpha$).

Projection operation reduces the signature of a given module. The relationships between the original concepts roles and individuals whose names remain in the signature are preserved (e.g. if $M \models \alpha$, $\alpha \in \mathcal{L}(\mathbf{S})$, then $\pi_{\mathbf{S}}(M) \models \alpha$).

Rename operation uses the notion of a *signature mapping.* Signature mapping γ is a triple of three functions $(\gamma_{\mathrm{C}}, \gamma_{\mathrm{R}}, \gamma_{\mathrm{I}})$ each of them being a bijection from $\mathcal{N}$ to $\mathcal{N}$. By $\gamma(\mathbf{S})$ we mean $\gamma_{\mathrm{C}}(\mathbf{AC}(\mathbf{S})) \uplus \gamma_{\mathrm{R}}(\mathbf{AR}(\mathbf{S})) \uplus \gamma_{\mathrm{I}}(\mathbf{Ind}(\mathbf{S}))$, and by $\gamma(\mathcal{I})$ where $\mathcal{I}$ is an interpretation we denote the interpretation $\mathcal{I}'$ such that $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ and $\gamma(X)^{\mathcal{I}'} = X^{\mathcal{I}}$ for every X. Rename preserves the relationships between concepts, roles, and individuals, however with respect to their name changes (e.g. if $M \models \alpha$, $\alpha \in \mathcal{L}(S(M))$, then $\rho_\gamma(M) \models \gamma(\alpha)$, where by $\gamma(\alpha)$ we understand the transformed α sentence in which all the names were systematically changed in accordance with γ).

For the second category of activities we introduce the operators of *union* ($\cup$), *intersection* ($\cap$) and *complement* ($\neg$) operating on the set of models.

$$\begin{array}{ll} M_1 \cup M_2 = (\mathbf{S}(M_1), \mathbf{W}(M_1) \cup \mathbf{W}(M_2)) & \mathbf{S}(M_1) = \mathbf{S}(M_2) \\ M_1 \cap M_2 = (\mathbf{S}(M_1), \mathbf{W}(M_1) \cap \mathbf{W}(M_2)) & \mathbf{S}(M_1) = \mathbf{S}(M_2) \\ \neg M = (\mathbf{S}(M), \mathfrak{I} - \mathbf{W}(M)) & \end{array}$$

Union, intersection and complement operations perform respective set-theoretic operations on sets of models of the s-modules. The condition that $\mathbf{S}(M_1) = \mathbf{S}(M_2)$ is not very restrictive because we can easily extend the binary operations to *generalized* ones. For example, generalized union $\cup_g$ can be defined as $M_1 \cup_g M_2 = \epsilon_{\mathbf{S}(M_2)}(M_1) \cup \epsilon_{\mathbf{S}(M_1)}(M_2)$.

Union and complement operators are (along with projection) non-linguistic (strictly semantic), i.e. their use may lead to generation of a s-module M for which there does not exist any corresponding set of sentences S in $\mathcal{L}$. This issue will be elaborated on in the further part of the paper.

The above definitions of s-module and operations do not depend on any particular logic. However, in the third category of operations we have to choose a language for relating terms. We have made the assumption that we use one of the standard Description Logics. Similarly like for RDBs, to relate two terms from different s-modules, an algebra user have to combine the two modules first (using union, intersection or difference) and then formulate DL sentences (e.g. axioms like $C \sqsubseteq D$) constraining the set of models to those which satisfy the sentences. To permit the user to do the latter we introduce the *selection* (σ) operation.

$$\sigma_\alpha(M) = (\mathbf{S}(M), \{\mathcal{I} \in \mathbf{W}(M) : \mathcal{I} \models \alpha\}) \qquad \alpha \in \mathcal{L}(\mathbf{S}(M))$$

Selection operation leaves in the module only these interpretations that are models of a chosen sentence α. Obviously $\sigma_\alpha(M) \models \alpha$.

It is easy to show that the outcome of all of the defined operations is a s-module itself (principally, it satisfies the condition $\mathbf{W}(M) = \mathbf{W}(M)|\mathbf{S}(M)$).

3.2 General properties of the algebra

In this section we present some basic algebraic laws for s-module algebra. One of the main results in this section is that Codd's relational algebra laws are also valid for s-module algebra. In order to transfer the laws we need to create a proper analogy between the two models. In this analogy a s-module is a counterpart to a single relation. The signature of a s-module is analogous to relation schema (i.e. names of the attributes). Consequently the set of models corresponds to a set of tuples contained in a relation. And since each axiom constrain the set of possible models, their influence is analogous to θ-restrictions, which (used in selection) reduce the number of tuples in a relation.

The analogy with RDBs is strong enough to ensure that relational algebra laws are also valid for s-module algebra. This result comes from the fact that both algebras are variants of cylindric algebras described in [5]. For relational algebra it has been shown in [6]. More precisely the authors of [6] show a correspondence between relational and diagonal-free cylindric algebra.

Let us recall the definition of this cylindric algebra [5]:

Definition 2. *A* diagonal free cylindric algebra of dimension ω *is a structure* $(A, +, \cdot, -, 0, 1, \mathsf{c}_i)$, $i \in [1..\omega)$, *such that* 0 *and* 1 *are distinguished elements of* A, $-$ *and* c_i *are unary operators on* A, $+$ *and* $\cdot$ *are binary operators on* A; *and the following conditions are satisfied for any* $x, y \in A$ *and any* $j, k \in [1..\omega)$*:*

$$\begin{aligned}
&(C_0) \quad (A, +, \cdot, -, 0, 1) \textit{ is a Boolean algebra}\\
&(C_1) \quad \mathsf{c}_j 0 = 0\\
&(C_2) \quad x \cdot \mathsf{c}_j x = x\\
&(C_3) \quad \mathsf{c}_j (x \cdot \mathsf{c}_j y) = \mathsf{c}_j x \cdot \mathsf{c}_j y\\
&(C_4) \quad \mathsf{c}_j \mathsf{c}_k x = \mathsf{c}_k \mathsf{c}_j x
\end{aligned}$$

We will show that s-module algebra is homomorphic to a diagonal free cylindric algebra of dimension ω. An issue we have to deal here with is rather technical and concerns module signature. We deal with this problem in the following way. We establish an equivalence relation $=_\updownarrow$ between modules such that $M_1 =_\updownarrow M_2$ iff $\mathbf{W}(M_1) = \mathbf{W}(M_2)$. We may identify the equivalence class $[M]_{=_\updownarrow}$ (we skip the $=_\updownarrow$ symbol henceforth, writing simply $[M]$) with the module $M^\uparrow = (\widehat{\mathbf{S}}, \mathbf{W}(M))$ or the module $M^\downarrow = (\mathbf{S}^\downarrow(M), \mathbf{W}(M))$ where by $\mathbf{S}^\downarrow(M)$ we mean the smallest signature $\mathbf{S}$ such that $\mathbf{W}(\pi_{\mathbf{S}}(M)) = \mathbf{W}(M)$. We then prove that the algebra over $\mathfrak{M}/ =_\updownarrow$ with slightly redefined operators satisfies the requirements (C_0) to (C_4).

Theorem 1. *Provided that* $|\widehat{\mathbf{S}}| = \omega$ *the structure* $(\mathfrak{M}/ =_\updownarrow, \cup_\updownarrow, \cap_\updownarrow, \neg_\updownarrow, [\check{M}], [\hat{M}], \mathsf{r}_X)$ *where* $X \in \widehat{\mathbf{S}}$ *and*

$$\begin{aligned}
\check{M} &= (\emptyset, \emptyset)\\
\hat{M} &= (\emptyset, \mathfrak{I})
\end{aligned}$$

$$\begin{aligned}
[M_1] \cup_{\updownarrow} [M_2] &= [M_1 \cup_g M_2]\\
[M_1] \cap_{\updownarrow} [M_2] &= [M_1 \cap_g M_2]\\
\neg_{\updownarrow} [M] &= [\neg M]\\
\mathsf{r}_X\, [M] &= \left[\pi_{\mathbf{S}(M)-\{X\}}(M)\right]
\end{aligned}$$

is a diagonal free cylindric algebra of dimension ω.

Proof (sketch). We show that conditions (C_1) to (C_4) are satisfied (M_1, M_2 are any modules from $\mathfrak{M}$; X, Y are any terms from $\widehat{\mathbf{S}}$):

$$\begin{aligned}
&(C_1)\ \ \mathsf{r}_X[\check{M}] = [\pi_{\mathbf{S}^{\downarrow}(M)-\{X\}}(\check{M})] = [\pi_{\emptyset}(\check{M})] = [\check{M}]\\
&(C_2)\ \ [M_1] \cap_{\updownarrow} \mathsf{r}_X[M_1] = [M_1] \cap_{\updownarrow} [\pi_{\mathbf{S}(M_1)-\{X\}}(M_1)] =\\
&\quad [M_1 \cap_g \pi_{\mathbf{S}^{\downarrow}(M_1)-\{X\}}(M_1)] = [M_1 \cap \epsilon_{\mathbf{S}(M_1)}(\pi_{\mathbf{S}(M_1)-\{X\}}(M_1))] =\\
&\quad [(\mathbf{S}(M_1), \mathbf{W}(M_1) \cap \mathbf{W}(M_1)|(\mathbf{S}^{\downarrow}(M_1) - \{X\}))] = [M_1]
\end{aligned}$$

For proving the next equality we will use the following fact[1] $(\mathbf{I} \cap \mathbf{J}|\mathbf{S})|\mathbf{S} = \mathbf{I}|\mathbf{S} \cap \mathbf{J}|\mathbf{S}$.

$$\begin{aligned}
&(C_3)\ \ \mathsf{r}_X([M_1] \cap_{\updownarrow} \mathsf{r}_X[M_2]) = \mathsf{r}_X([M_1^{\uparrow}] \cap_{\updownarrow} \mathsf{r}_X[M_2^{\uparrow}]) =\\
&\quad [\pi_{\widehat{\mathbf{S}}-\{X\}}(M_1^{\uparrow} \cap_g \pi_{\widehat{\mathbf{S}}-\{X\}}(M_2^{\uparrow}))] =\\
&\quad [(\widehat{\mathbf{S}} - \{X\}, (\mathbf{W}(M_1) \cap \mathbf{W}(M_2)|(\widehat{\mathbf{S}} - \{X\}))|(\widehat{\mathbf{S}} - \{X\}))] =\\
&\quad [(\widehat{\mathbf{S}} - \{X\}, \mathbf{W}(M_1)|(\widehat{\mathbf{S}} - \{X\}) \cap \mathbf{W}(M_2)|(\widehat{\mathbf{S}} - \{X\}))] =\\
&\quad [\pi_{\widehat{\mathbf{S}}-\{X\}}(M_1^{\uparrow}) \cap_g \pi_{\widehat{\mathbf{S}}-\{X\}}(M_2^{\uparrow})] = \mathsf{r}_X[M_1] \cap_{\updownarrow} \mathsf{r}_X[M_2]
\end{aligned}$$

To prove (C_4) we will take advantage of the equality: $\mathbf{I}|\mathbf{S}_1|\mathbf{S}_2 = \mathbf{I}|\mathbf{S}_2|\mathbf{S}_1$:

$$\begin{aligned}
&(C_4)\ \ \mathsf{r}_X\mathsf{r}_Y[M_1] = [\pi_{\widehat{\mathbf{S}}-\{X\}}(\pi_{\widehat{\mathbf{S}}-\{Y\}}(M_1^{\uparrow})^{\uparrow})^{\uparrow}] =\\
&\quad [(\widehat{\mathbf{S}}, \mathbf{W}(M_1)|(\widehat{\mathbf{S}} - \{Y\})|(\widehat{\mathbf{S}} - \{X\}))] =\\
&\quad [(\widehat{\mathbf{S}}, \mathbf{W}(M_1)|(\widehat{\mathbf{S}} - \{X\})|(\widehat{\mathbf{S}} - \{Y\}))] =\\
&\quad [\pi_{\widehat{\mathbf{S}}-\{Y\}}(\pi_{\widehat{\mathbf{S}}-\{X\}}(M_1^{\uparrow})^{\uparrow})^{\uparrow}] = \mathsf{r}_Y\mathsf{r}_X[M_1]
\end{aligned}$$

The full proof contains the step showing that operations $\cup_{\updownarrow}, \cap_{\updownarrow}, \neg_{\updownarrow}$ are properly defined and shows that the condition (C_0) is also satisfied.

The theorem 1 is very strong and allows us to transfer all laws of cylindric (and relational) algebra to s-module algebra. The following proposition collects some of them.

Proposition 1. *The following laws hold for s-module algebra:*

1. *For union, intersection and negation standard set-theoretic laws hold (e.g. union and intersection are idempotent, associative and commutative).*

[1] On the basis of the fact that the relation $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$ is associative: $\mathcal{I}_0 \in ((\mathbf{I} \cap \mathbf{J}|\mathbf{S})|\mathbf{S}) \Leftrightarrow \exists \mathcal{I} \in \mathbf{I} : \exists \mathcal{J} \in \mathbf{J} : \mathcal{I}_0|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}} \Leftrightarrow \mathcal{I}_0 \in (\mathbf{I}|\mathbf{S} \cap \mathbf{J}|\mathbf{S})$.

2. For projection we have:
   a. $\pi_{\mathbf{S}}(M) = \pi_{\mathbf{S}}(\pi_{\mathbf{S}}(M))$,
   b. $\pi_{\mathbf{S}(M)}(M) = M$
   c. $\pi_{\mathbf{S}}(\pi_{\mathbf{S}'}(M)) = \pi_{\mathbf{S}\cap\mathbf{S}'}(M) = \pi_{\mathbf{S}'}(\pi_{\mathbf{S}}(M))$,
   d. $\pi_{\mathbf{S}}(\neg M) = \neg\pi_{\mathbf{S}}(M)$,
   e. $\pi_{\mathbf{S}}(M_1 \cup M_2) = \pi_{\mathbf{S}}(M_1) \cup \pi_{\mathbf{S}}(M_2)$,
   f. $\pi_{\mathbf{S}}(M_1 \cap_g \pi_{\mathbf{S}}(M_2)) = \pi_{\mathbf{S}}(M_1) \cap \pi_{\mathbf{S}}(M_2)$
3. For selection we have:
   a. $\sigma_\alpha(M) = \sigma_\alpha(\sigma_\alpha(M))$,
   b. $\sigma_\alpha(\sigma_\beta(M)) = \sigma_\beta(\sigma_\alpha(M))$,
   c. $\sigma_\alpha(M_1 \cup M_2) = \sigma_\alpha(M_1) \cup \sigma_\alpha(M_2)$,
   d. $\sigma_\alpha(M_1 \cap M_2) = \sigma_\alpha(M_1) \cap \sigma_\alpha(M_2)$

4 Discussion of the Results

By proposing the s-module algebra we do not claim that ontology engineers should abandon sentences and focus solely on models and interpretations. On the contrary, we are convinced that linguistic representations (based on sentences but not necessarily just sets of sentences) are most appropriate for the majority of engineering tasks. In our intention s-module algebra plays a role of a point of reference for attempts of creating proper representations of semantics of modules and, perhaps, proper algebras for manipulating these representations.

Naturally to approve the s-module algebra in the role of such yardstick we should first agree that the range of offered operations is complete, i.e. suitable for describing some set of reference problems. The issue is at least to some extent subjective, but we have an important argument supporting the proposed range.

In the previous part of the paper we showed the strong correspondence between s-module algebra and Codd's relational algebra. Although there is no guarantee that operators used in RDBs are relevant for ontology modules, this analogy gives us the argument in justifying the proposed range of operators. "Completeness" (proper choice of operators) for the relational algebra was shown by Codd with a proof that with algebraic constructions we can express every set of tuples satisfying a certain expression formulated in a fragment of FOL called relational calculus [2]. Because it is essentially a feature of cylindric algebra the s-module algebra has the similar property.

One of the reasons why already created "syntactic" algebras are not, in our intuition, proper for the role of reference point is the fact that in these approaches straightforward definitions often fail to capture the intended semantic meaning. For example, NeOn algebra characterize each module M by the set of sentences satisfied by it (simplifying a bit and not delving into the details we may denote the set by $\{\alpha \in \mathcal{L}(\widehat{\mathbf{S}}) : M \models \alpha\}$). The most intuitive way of defining the difference of two modules (and the one proposed in [1]) $M_1 - M_2$ is to say that this is the module M' which satisfies α iff $M_1 \models \alpha$ and not $M_2 \models \alpha$. Because M_2 satisfies also all tautologies (e.g. $\top \equiv \top$) it follows that M' cannot satisfy any tautology, which is impossible. While there are ways of improving the definition given in the above example, we believe that problems of this type are hardly avoidable and to some extent inherent for "syntactic" (or "deductive") approaches. In contrast, union, intersection, and negation in s-module algebra, defined in an intuitive way, naturally form a Boolean algebra.

One problem we have to bear in mind is that simple sets of sentences may not give enough expressive power to cover the whole range of s-module algebra operations. Indeed this is the case, and to illustrate it we formalize the issue.

Definition 3 ($\mathcal{L}$-representable s-module). *We call an s-module M $\mathcal{L}$-*representable *iff there exists a set of sentences $S \subseteq \mathcal{L}(\mathbf{S}(M))$ such that $\mathbf{W}(M) = \{\mathcal{I} \in \mathfrak{I} : \forall\alpha \in S : \mathcal{I} \models \alpha\}$.*

In other words $\mathcal{L}$-representable s-modules are the modules whose models are all the models of a particular set of sentences from $\mathcal{L}(\mathbf{S}(M))$. We may denote a specific $\mathcal{L}$-representable s-module as $M(\mathbf{S}, S)$ where $S \subset \mathcal{L}(\mathbf{S})$, or simply $M(S)$, if module signature consists of all term used in sentences from S, e.g. $M(\{A \sqsubseteq B, A(a)\})$ has the signature $\{A, B, a\}$.

While the representation in the form of sentences is convenient, in general we cannot represent every s-module with a set of DL sentences. Specific algebra operations (namely union, complement and projection) can give the outcome which is not $\mathcal{L}$-representable even if all operands are. The example below generalizes even to very expressive logics like $\mathcal{SHOIQ}$.

Example 1 (non-representable s-module). Consider a union M of two $\mathcal{ALC}$-representable s-modules $M = M(\{A \sqsubseteq B\}) \cup M(\{B \sqsubseteq A\})$. Notice that neither $M \models A \sqsubseteq B$ nor $M \models B \sqsubseteq A$ is true. However if we perform the intersection: $M' = M \cap_g M(\{A \sqcap \neg B(a), \neg A \sqcap B(b)\})$ we will obtain a s-module with no models $\mathbf{W}(M') = \emptyset$.

The fact that s-module algebra offers greater expressiveness than sets of sentences stems the problem of transferring the practical results obtained with use of the algebra to description of practical problems which demand finite and often linguistic representations. This means that for the transfer of the results we have to use some "translation" of obtained results to the desired representation. The kind of "translation" should be appropriate to the problem we want to solve.

One of the possible methods we may consider is *rounding*. By rounding we mean reducing or expanding the sets of models to an $\mathcal{L}$-representable one. An example of rounding function is $M^* = (\mathbf{S}, \{\mathcal{I} \in \mathfrak{I} : \forall\alpha \in \mathcal{L}(\mathbf{S}(M)) : \mathcal{I} \models \alpha \Leftrightarrow \mathbf{W} \models \alpha\})$. We have to remember, however, that by using rounding we leave the world of purely semantic operations. Rounding requires special care with use of algebra operations (e.g. in general it is not true that $M_1^* \cap M_2^* = (M_1 \cap M_2)^*$) and it shows its usefulness in situations when we have to describe an issue which is strictly connected with "syntactic" or "deductive" approach.

To illustrate possible use of rounding, below we show an example of use s-module algebra for describing problems connected with important and recently widely discussed subject of safety exploiting the notion of deductive conservative extension ([7], [4]).

Firstly, let us recall the definition of model conservative extension (by an ontology $\mathcal{O}$ we understand a set of sentences, $\mathsf{Sig}(\mathcal{O})$ is a signature containing all terms used in sentences of $\mathcal{O}$; we assume that we use certain DL $\mathcal{L}$):

Definition 4 (conservative extensions [4]). *Given two ontologies $\mathcal{O}$ and $\mathcal{O}_1 \subseteq \mathcal{O}$, we say that $\mathcal{O}$ is a* model $\mathbf{S}$-conservative extension *of $\mathcal{O}_1$, if for every model $\mathcal{I}$ of $\mathcal{O}_1$, there exists a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$. We say that $\mathcal{O}$ is a* deductive $\mathbf{S}$-conservative extension *of $\mathcal{O}_1$ if for every sentence $\alpha \in \mathcal{L}(\mathbf{S})$ we have $\mathcal{O} \models \alpha$ iff $\mathcal{O}_1 \models \alpha$.*

Having in mind that $\mathbf{I}|\mathbf{S} = \{\mathcal{J} \in \mathfrak{I} : \exists \mathcal{I} \in \mathbf{I} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}\}$ (cf. Preliminaries), it is straightforward to show that, given two ontologies $\mathcal{O}$ and $\mathcal{O}_1 \subseteq \mathcal{O}$, $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ iff $\{\mathcal{I} : \mathcal{I} \models \mathcal{O}\}|\mathbf{S} = \{\mathcal{I} : \mathcal{I} \models \mathcal{O}_1\}|\mathbf{S}$. The following proposition is an immediate corollary:

Proposition 2. *Given two ontologies $\mathcal{O}$ and $\mathcal{O}_1 \subseteq \mathcal{O}$: $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ iff $\pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O}))) = \pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O}_1)))$; $\mathcal{O}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ iff $\pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O})))^* = \pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O}_1)))^*$.*

Note that in the above proposition we used rounding to describe a "deductive" notion. Naturally it is also immediate to state that $\pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O})) = \pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O}_1))) \Rightarrow \pi_{\mathbf{S}}(\epsilon_{\mathbf{S}} M(\mathcal{O}))^* = \pi_{\mathbf{S}}(\epsilon_{\mathbf{S}}(M(\mathcal{O}_1)))^*$.

The authors of [4] define an important notion of *safety for a signature*:

Definition 5 (safety for a signature [4]). *We say that ontology $\mathcal{O}$ is* safe for a signature $\mathbf{S}$ *if for every ontology $\mathcal{O}'$ with $\mathsf{Sig}(\mathcal{O}) \cup \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$ we have that $\mathcal{O} \cap \mathcal{O}'$ is a $\mathsf{Sig}(\mathcal{O}')$-deductive conservative extension of $\mathcal{O}'$.*

This notion is important as it describes an ontology which can safely import any ontology $\mathcal{O}'$ that shares with it only symbols from $\mathbf{S}$. [4] also provides the sufficient condition for the safety:

Proposition 3 (safety for a signature vs. model conservative extension [4]). *Let $\mathcal{O}$ be an ontology such that $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of the empty ontology $\mathcal{O}_1 = \emptyset$. Then $\mathcal{O}$ is safe for $\mathbf{S}$.*

We will prove this proposition with use of s-module algebra. To do this, according to Definition 5 and Proposition 2 it is sufficient to prove that for any ontology $\mathcal{O}'$ such that $\mathsf{Sig}(\mathcal{O}) \cup \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$ ($\natural$) the following is true: $\pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}')) = \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}) \cap_g M(\mathcal{O}'))$. From the assumption we have $\pi_{\mathbf{S}}(M(\mathcal{O})) = \pi_{\mathbf{S}}(M(\emptyset))$, so, using ($\natural$), we can state that $\pi_{\mathsf{Sig}(\mathcal{O})\cap\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O})) = \pi_{\mathsf{Sig}(\mathcal{O})\cap\mathsf{Sig}(\mathcal{O}')}(M(\emptyset))$ ($\flat$). Having this stated, the proof is straightforward:

$$\begin{aligned}
&\pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}) \cap_g M(\mathcal{O}')) && \\
&= \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}) \cap_g \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}'))) && \text{(by Th. 1 p. 2.b)} \\
&= \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O})) \cap_g \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}')) && \text{(by Th. 1 p. 2.f)} \\
&= \pi_{\mathsf{Sig}(\mathcal{O}')}(\pi_{\mathsf{Sig}(\mathcal{O})}(M(\mathcal{O}))) \cap_g \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}')) && \text{(by Th. 1 p. 2.b)} \\
&= \pi_{\mathsf{Sig}(\mathcal{O}')\cap\mathsf{Sig}(\mathcal{O})}(M(\mathcal{O})) \cap_g \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}')) && \text{(by Th. 1 p. 2.c)} \\
&= \pi_{\mathsf{Sig}(\mathcal{O})\cap\mathsf{Sig}(\mathcal{O}')}(M(\emptyset)) \cap_g \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}')) && \text{(by } (\flat)) \\
&= \pi_{\mathsf{Sig}(\mathcal{O}')}(M(\mathcal{O}')) && (\hat{M} \cap_g M = M)
\end{aligned}$$

Although rounding makes each s-module $\mathcal{L}$-representable, we may consider using different linguistic representations for other purposes. During our study of decidability problems for s-module algebra we came to the result that representation of modules in the form of sets of sentences can be advantageous. The following definition formalizes the idea.

Definition 6 ($\mathcal{L}$-(2)-representability). *We call an s-module M* $\mathcal{L}$-(2)-representable *iff there exists a finite set $S = \{S_1 \dots S_k\}$ of sets of sentences $S_i \subseteq \mathcal{L}(\mathbf{S}(M))$ such that $\mathbf{W}(M) = \{\mathcal{I} : \exists i \in [1..k] : \mathcal{I} \models S_i\}$. We call every such S* (2)-representation *of M. We call the logic $\mathcal{L}$* (2)-representable wrt. a set of operators A *iff every s-module being an outcome of algebraic operations from* A *on $\mathcal{L}$-representable s-modules is $\mathcal{L}$-(2)-representable.*

(2)-representability is useful because having a (2)-representation of an s-module we can easily check satisfiability of concept C against the s-module (check if there exists an interpretation $\mathcal{I} \in \mathbf{W}(M)$ for which $C^{\mathcal{I}} \neq \emptyset$). The procedure consist simply of checking its satisfiability against each set of sentences.[2]

Proposition 4. *$\mathcal{SHOIQ}$ is (2)-representable wrt.* $\{\epsilon, \rho, \cup, \cap, \sigma\}$

Proof (sketch). We associate each s-module with sets of sentences (denoted $M \triangleright \{S_1 \dots S_k\}$) and prove $\{S_1 \dots S_k\}$ is its (2)-representation (claim ($\flat$)). The proof goes by structural induction over the tree of algebraic expressions. Representable modules of the form $M(S)$ are associated with S ($M \triangleright \{S\}$), and for these modules ($\flat$) obviously holds (which forms the induction basis). Each algebraic operation produces a new module associated with newly constructed sets:

$$\begin{aligned}
\epsilon(M \triangleright \{S_1 \dots S_k\}) &= M' \triangleright \{S_1 \dots S_k\} \\
\rho_\gamma(M \triangleright \{S_1 \dots S_k\}) &= M' \triangleright \{\gamma(S_i) : i \in [1..k]\} \\
\sigma_\alpha(M \triangleright \{S_1 \dots S_k\}) &= M' \triangleright \{S_i \cup \{\alpha\} : i \in [1..k]\} \\
M_1 \triangleright \{S_1^1 \dots S_k^1\} \cup M_2 \triangleright \{S_1^2 \dots S_l^2\} &= M' \triangleright (\{S_1^1 \dots S_k^1\} \cup \{S_1^2 \dots S_l^2\}) \\
M_1 \triangleright \{S_1^1 \dots S_k^1\} \cap M_2 \triangleright \{S_1^2 \dots S_l^2\} &= M' \triangleright \{S_i^1 \cup S_j^2 : i \in [1..k], j \in [1..l]\}
\end{aligned}$$

It is straightforward to show that for all the operators (under the induction hypothesis) claim ($\flat$) holds.

Generally, (2)-representations are not strong enough to represent the outcome of projection (π). The reason for that is the fact that projection $\pi_{\mathbf{S}}(M)$ may constrain the signature of the module in the way that some of the sentences may fall outside $\mathcal{L}(\mathbf{S})$.[3] Nevertheless, we have to remember these sentences somehow, as they may imply important relationships between terms in the remaining part of the signature. To overcome this problem we may separate from $\widehat{\mathbf{S}}$ a special

[2] It is worth to note that every $\mathcal{L}$-(2)-representable module is $\mathcal{L}'$-representable if $\mathcal{L}'$ is the extension of $\mathcal{L}$ by disjunctions between sets of sentences (e.g. similarly as proposed in [9]).

[3] Example of which can be found in [10] (Lemma 6).

subset $\mathbf{D}$ of "dummy" names and assume that they cannot be a part of s-module signature but may be used in the sentences in its representation. The following definition formally shows the idea.

Definition 7 ($\mathcal{L}$-(2)-$\mathbf{D}$-representability). *We call an s-module M* $\mathcal{L}$-(2)-$\mathbf{D}$-representable *iff there exists a finite set $S = \{S_1 \ldots S_k\}$ of sets of sentences $S_i \subseteq \mathcal{L}(\mathbf{S}(M) \cup \mathbf{D})$ such that $\mathbf{W}(M) = \{\mathcal{I} : \exists i \in [1..k] : \mathcal{I} \models S_i\}|\mathbf{S}(M)$. We call every such S* (2)-$\mathbf{D}$-representation *of M. We call the logic $\mathcal{L}$* (2)-$\mathbf{D}$-representable wrt. a set of operators A *iff every s-module being an outcome of algebraic operations from* A *on $\mathcal{L}$-representable s-modules is $\mathcal{L}$-(2)-$\mathbf{D}$-representable.*

Proposition 5. *$\mathcal{SHOIQ}$ is (2)-$\mathbf{D}$-representable wrt. $\{\epsilon, \pi, \rho, \cup, \cap, \sigma\}$*

Proof (sketch). We extend the proof of Proposition 4 by: $\pi_{\mathbf{S}}(M \triangleright \{S_1 \ldots S_k\}) = M' \triangleright \{\gamma_{\mathbf{S}}^{\mathbf{S}(M)}(S_i) : i \in [1..k]\}\}$ Special rename function $\gamma_{\mathbf{S}}^{\mathbf{S}'}$ changes the names from $\mathbf{S}'$ that fall outside $\mathbf{S}$ to unique "dummy" names from $\mathbf{D}$.

Use of dummy names allows us also to handle negation, as shown in the following proposition.

Proposition 6. *$\mathcal{SHOIQ}$ is (2)-$\mathbf{D}$-representable wrt. $\{\epsilon, \rho, \cup, \cap, \neg, \sigma\}$*

Proof (sketch). We use the same technique as in the proof of Proposition 4. For complement ($\neg$) we use the fact that the interpretation not being a model of $M \triangleright \{S_1 \ldots S_k\}$ has to contradict at least one sentence from each of the sets $\{S_1 \ldots S_k\}$. Thus we build a list $\{N_i\}$ of sets of sentences containing one negated sentence from each of the sets $\{S_1 \ldots S_k\}$, each $N_i = \tau_\neg(\{\alpha_1, \alpha_2, \ldots, \alpha_k\}), \alpha_j \in S_j$. We negate the sentences using $\tau_\neg$ translation as follows: $C \sqsubseteq D$ we transform to $\{x\} \sqsubseteq C \sqcap \neg D$; $R \sqsubseteq S$ to $\{x\} \sqsubseteq \exists R.\{y\}$, $\{x\} \sqsubseteq \neg\exists S.\{y\}$; $\mathsf{Trans}(R)$ to $\{x\} \sqsubseteq \exists R.\{y\}$, $\{y\} \sqsubseteq \exists R.\{z\}$, $\{x\} \sqsubseteq \neg\exists R.\{z\}$; x, y, z are dummy names. The resulting module is $M' \triangleright \{N_1 \ldots N_l\}$, where l is the number of sets N_i obtained with use of the described method.

5 Summary

In this paper we described an algebra, called s-module algebra, operating on semantically defined modules. We showed that this algebra is homomorphic to a diagonal free cylindric algebra and, as a consequence, to Codd's relational algebra. Basing on this homomorphism we proved some properties of operators defined for s-module algebra. We also showed some examples of how to use the algebra to describe selected known problems, e.g. satisfiability in non-linguistic s-modules or semantic and deductive conservative extension.

For future work we are planning to investigate the wider range of problems the proposed algebra can be useful to solve. Basing on it we also would like to derive and investigate algebras, operating on various linguistic representations, and study their usefulness in covering different tasks.

Acknowledgements. This work was partially supported by Polish Ministry of Research and Higher Education, project No. N N516 4115 33.

References

1. d'Aquin M. et al.: *NeOn Formalisms for Modularization: Syntax, Semantics, Algebra.* Neon Project, Deliverable D1.1.3, 2008.
2. Codd E. F.: *Relational Completeness of Data Base Sublanguages.* In: R. Rustin (ed.): *Database Systems*, pp. 65–98, Prentice Hall, also IBM Research Report RJ 987, San Jose, California, 1972.
3. Hall P., Hitchcock P., Todd S.: *An algebra of relations for machine computation.* In *Proceedings of the 2nd ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pp. 225–232, Palo Alto, California, 1975.
4. Grau B. C., Horrocks I., Kazakov Y., Sattler U.: *Modular Reuse of Ontologies: Theory and Practice.* In: *J. of Artificial Intelligence Research* (JAIR), Vol. 31, pp. 273–318, 2008.
5. Henkin, L., Monk J. D., Tarski A. *Cylindric Algebras pt. 1*, Studies in Logic and the Foundations of Mathematics, Vol. 64, North-Holland, Amsterdam, 1971.
6. Imielinski T., Lipski W. J.: *The Relational Model of Data and Cylindric Algebras.* In: *J. Comput. Syst. Sci.*, Vol. 28, No. 1, pp. 80–102, 1984.
7. Lutz C., Walther D., Wolter F.: *Conservative extensions in expressive description logics.* In: *Proc. of IJCAI-2007*, pp. 453–459, 2007.
8. Mitra P., Wiederhold G.: *An Ontology-Composition Algebra.* In: *Handbook on Ontologies*, pp. 171–216, Germany, 2004.
9. Konev, B., Lutz, C., Walther, D., Wolter, F.: *Formal Properties of Modularisation.* In: *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, LNCS 5445, Springer, 2009.
10. Konev, B., Walther, D., Wolter, F.: *The Logical Difference Problem for Description Logic Terminologies.* In: *Automated Reasoning, Proc. of IJCAR-2008*, LNCS 5195, pp. 259–274, Springer, 2008.