STUDIA INFORMATICA 2009
Volume 30 Number 2A (83)

Krzysztof GOCZYŁA, Aleksander WALOSZEK, Wojciech WALOSZEK
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki

ALGEBRA KONGLOMERATÓW JAKO NARZĘDZIE OPISU PROBLEMÓW PRZETWARZANIA ONTOLOGII[1]

Streszczenie. W niniejszym artykule zaprezentowano nowe narzędzie przetwarzania modularnych baz wiedzy – algebrę konglomeratów. Przedstawioną propozycję wyróżnia fakt, że traktujemy moduły bazy wiedzy *semantycznie*. W artykule zaprezentowano doświadczenia związane z wykorzystaniem algebry (m.in. do dekompozycji bazy wiedzy oraz jako wsparcie procesu wnioskowania) oraz naszkicowano kierunki jej przyszłego rozwoju.

Słowa kluczowe: ontologie, bazy wiedzy, modularyzacja

CONGLOMERATION ALGEBRA AS A TOOL FOR DESCRIBING ONTOLOGY PROCESSING PROBLEMS

Summary. In the paper we present a novel conglomeration algebra which is a basic tool in our approach to knowledge base modularization. In the algebra modules are treated *semantically*. The paper presents our experiences with using conglomeration algebra and sketch the directions of its further development.

Keywords: ontology, knowledge base, modularity

1. Wprowadzenie

Kolejne lata przynoszą wzrost zainteresowania zagadnieniem modularyzacji baz wiedzy opartych na logice opisowej [1] (ang. *Description Logics*, DL). Głównym czynnikiem motywującym jest tu nadzieja osiągnięcia dojrzałości procesu tworzenia ontologii oraz ich ponownego wykorzystania porównywalnej z tą, jaką osiągnęła inżynieria oprogramowania.

[1] Praca naukowa finansowana ze środków na naukę w latach 2007-2009 jako projekt badawczy MNiSzW.

Podczas prac nad systemem zarządzania bazami wiedzy opartymi na logice opisowej RKaSeA (prowadzonych przez grupę KMG w Katedrze Inżynierii Oprogramowania na Wydziale Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej), grupa projektowa stanęła przed zagadnieniem wyboru odpowiedniej metody modularyzacji. Ważnym czynnikiem była tu chęć wykorzystania potencjału modularyzacji w szerszym stopniu niż w innych znanych systemach. Jednym z celów było udostępnienie użytkownikowi możliwości tworzenia i przetwarzania modularnych baz wiedzy, ale także pozwolenie mu na wykonywanie bardziej zaawansowanych operacji, takich jak ekstrakcja fragmentu modułu, połączenie tego fragmentu z fragmentem innego modułu itp. Dodatkowym celem było także wykorzystanie metod automatycznej (i niewidocznej dla użytkownika) dekompozycji bazy wiedzy w celu przyspieszenia procesu wnioskowania.

W efekcie, również ze względu na brak uznanego pojedynczego standardu i formalizmu modularnych baz wiedzy, w ramach grupy opracowano własny formalizm, oparty na doświadczeniach związanych z pracami nad modelem SIM [14] i opisujący operacje na modułach bazy wiedzy. Formalizm jest stosunkowo prosty, jednak wystarczająco ekspresywny, aby opisać skomplikowane działania związane z dekompozycją i łączeniem modułów. Skupiono się tu silnie na semantycznym aspekcie modułów (tj. nie na zdaniach opisujących zawartość danego modułu, a na przestrzeni jego możliwych interpretacji). Z uwagi na to, że przy takim podejściu moduł traci strukturę zdaniową (brak już tutaj np. typowego podziału na TBox i ABox), formalizm ten nazwano algebrą konglomeratów (gdzie *konglomerat* to semantyczna postać modułu). Algebra ta wzorowana jest na algebrze relacji Codda [2].

W dalszej części opracowania wprowadzono podstawowe pojęcia algebry konglomeratów i pokazano przykłady ich wykorzystania. Przeprowadzono także dyskusję zagadnień związanych z rozstrzygalnością oraz porównano przedstawioną propozycję z istniejącymi rozwiązaniami w zakresie modularyzacji. Artykuł kończy krótka dyskusja na temat możliwych kierunków rozwoju proponowanej metody.

2. Zagadnienia wstępne

W niniejszym artykule pojęć ontologii i bazy wiedzy używa się zamiennie, w znaczeniu określonym przez DL. W dalszej części artykułu wykorzystano pojęcie *sygnatury*[2] (słownika), oznaczanej jako $\mathbf{S} = \mathbf{C} \uplus \mathbf{R} \uplus \mathbf{I}$ i przyjęto, że dla każdego dialektu logiki opisowej $\mathcal{L}$ sygnatura jest sumą rozłączną zbiorów *nazw konceptów* (**C**), *nazw ról* (**R**) oraz *nazw osobników* (**I**). Poszczególne części danej sygnatury **S** oznaczane są odpowiednio

[2] Pojęcia tego nie należy mylić z sygnaturą binarną wprowadzoną w innych naszych pracach.

wyrażeniami **C**(**S**), **R**(**S**), **I**(**S**). Przyjęto także, że istnieje pewien z góry określony zbiór wszystkich możliwych nazw, oznaczony $\mathcal{N}$ (**C**, **R**, **I** $\subseteq \mathcal{N}$).

Wybrany dialekt logiki opisowej $\mathcal{L}$ wyznacza zbiory (również złożonych) konceptów, ról i osobników, które można skonstruować wykorzystując operatory $\mathcal{L}$ oraz nazwy z sygnatury **S** (oznaczane odpowiednio przez $\mathcal{L}_C(\mathbf{S})$, $\mathcal{L}_R(\mathbf{S})$, $\mathcal{L}_I(\mathbf{S})$). Na przykład, jeżeli wybranym dialektem $\mathcal{L}$ jest $\mathcal{ALC}$, to $\mathcal{L}_C(\mathbf{S}) ::= A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C \mid \top \mid \bot$, gdzie $A \in \mathbf{C}(\mathbf{S})$, $C, D \in \mathcal{L}_C(\mathbf{S})$, $R \in \mathcal{L}_R(\mathbf{S})$.

S-*interpretacją* (lub krócej: interpretacją, gdy sygnatura **S** jest znana z kontekstu lub dowolna) $\mathcal{I}$ nazywamy parę $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$, gdzie $\Delta^\mathcal{I}$ jest niepustym zbiorem nazywanym *dziedziną* interpretacji, a $\cdot^\mathcal{I}$ jest *funkcją interpretacyjną* przypisującą każdemu konceptowi atomowemu $A \in \mathbf{C}(\mathbf{S})$ podzbiór dziedziny $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, każdej roli atomowej $R \in \mathbf{R}(\mathbf{S})$ binarną relację $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, a każdemu osobnikowi $a \in \mathbf{I}(\mathbf{S})$ element dziedziny $a^\mathcal{I} \in \Delta^\mathcal{I}$. Zarazem każdy dialekt logiki opisowej $\mathcal{L}$ definiuje zasady rozszerzania interpretacji bazowych terminów na terminy złożone, np. w $\mathcal{ALC}$ $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$. Interpretacja pozwala więc na nadanie znaczenia poszczególnym elementom wykorzystywanego słownika.

Każdy dialekt logiki opisowej pozwala na formułowanie aksjomatów i asercji (zdań) oraz definiuje warunki *spełniania* danego zdania przez **S**-interpretację $\mathcal{I}$. Zbiór zdań danego dialektu $\mathcal{L}$ wykorzystujący nazwy z sygnatury **S** oznaczamy jako $\mathcal{L}(\mathbf{S})$. Fakt, że interpretacja $\mathcal{I}$ spełnia dane zdanie $\alpha \in \mathcal{L}(\mathbf{S})$ (*jest jego modelem*), oznaczamy przez $\mathcal{I} \vDash \alpha$.

3. Algebra konglomeratów

3.1. Podstawowe definicje

W prezentowanej pracy przyjęto stanowisko, że użytkownik zainteresowany jest wnioskami płynącymi z danego modułu (fragmentu bazy wiedzy), nie interesuje go natomiast forma zdań, które zostały wykorzystane, aby dany moduł utworzyć. W konsekwencji tego założenia zdefiniowano pojęcie konglomeratu (czasem zamiennie w mniej formalnych kontekstach używamy słowa moduł) ściśle semantycznie, skupiając się wyłącznie na jego interpretacjach.

Definicja 1. *Konglomeratem M* nazywamy parę (**S**, **W**) składającą się z sygnatury **S** (*sygnatury* konglomeratu) i zbioru **W** zawierającego **S**-interpretacje. Kiedy chcemy się odwołać do jednego z elementów pary, używamy notacji odpowiednio **S**(M) oraz **W**(M). Każdą **S**-interpretację z klasy **W** nazywamy *modelem* konglomeratu M.

Zgodnie z tą definicją każdy konglomerat po prostu zawiera wszystkie swoje modele. Mówimy, że dany konglomerat M spełnia zdanie α, co oznaczamy $M \vDash \alpha$, wtwg.

$\forall \mathcal{I} \in \mathbf{W}(M)$: $\mathcal{I} \models \alpha$. W praktyce przyjmuje się, że wszystkie dziedziny interpretacji z $\mathbf{W}$ są podzbiorami pewnego zadanego zbioru Δ.

U podstaw proponowanego formalizmu leży założenie, że sposób modularyzacji ontologii wprowadzony przez jej autorów nie jest jedynym możliwym i może czasem zaistnieć konieczność jego zmodyfikowania dla zaspokojenia konkretnej potrzeby użytkownika. Ujmując to nieco innymi słowy: twórcy ontologii nie są w stanie przewidzieć wszystkich możliwości jej ponownego użycia. Z konieczności skupiają się na pewnym konkretnym kontekście (lub, w najlepszym przypadku, pewnym zbiorze kontekstów) jej wykorzystania, a kontekst ten może okazać się niewłaściwy dla określonego zastosowania. Właśnie ze względu na to założenie algebra konglomeratów została skonstruowana w taki sposób, by dostarczyć użytkownikowi szerokiej gamy narzędzi do manipulacji zawartością poszczególnych konglomeratów (w tym ich dekompozycji i łączenia). Narzędzia te operują na obu częściach konglomeratu: sygnaturach i modelach.

3.2. Działania algebry konglomeratów

W tym podrozdziale zdefiniowano działania (lub operacje, w dalszej części artykułu pojęcia te używane są zamiennie) algebry konglomeratów.

W poniższych definicjach zakładamy, że wybrane są dialekt logiki opisowej $\mathcal{L}$ oraz zbiór Δ będący nadzbiorem wszystkich dziedzin wszystkich modeli konglomeratów. Oznaczamy zbiór wszystkich konglomeratów jako $\mathfrak{M}$, zbiór wszystkich sygnatur jako $\mathfrak{S}$, a zbiór wszystkich $\mathbf{S}$-interpretacji jako $\mathfrak{I}(\mathbf{S})$. Wykorzystujemy również pojęcie *obcięcia* interpretacji. $\mathcal{I}'$ jest obcięciem $\mathbf{S}$-interpretacji $\mathcal{I}$ do sygnatury $\mathbf{S}'$ (co zapisujemy jako $\mathcal{I}' = \mathcal{I}|\mathbf{S}'$, zakładając przy tym, że $\mathbf{S}' \subseteq \mathbf{S}$) wtwg. $\mathcal{I}'$ jest $\mathbf{S}'$-interpretacją, dla której spełnione są następujące warunki: $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ oraz $X^{\mathcal{I}'} = X^{\mathcal{I}}$ dla każdego $X \in \mathbf{S}'$. W opisach działań unarnych oznaczamy argument jako M. Argumenty operacji binarnych oznaczamy jako M_1 i M_2 (ich sygnatury i zbiory modeli odpowiednio jako $\mathbf{S}_1$, $\mathbf{S}_2$ i $\mathbf{W}_1$, $\mathbf{W}_2$), a symbole operacji binarnych zapisujemy w notacji infiksowej. M' zawsze oznacza wynik.

Podstawowe operacje działające na słowniku to komplementarne działania **rozszerzenia sygnatury** ($\varepsilon_{\mathbf{S}}$) i **projekcji** ($\pi_{\mathbf{S}}$). Rozszerzenie sygnatury poszerza zakres stosowanego w danym konglomeracie słownika o terminy z zadanego zbioru $\mathbf{S}$. Dopuszczalny zbiór interpretacji każdego istniejącego wcześniej terminu musi być zachowany, podobnie jak zależności pomiędzy konceptami, rolami i osobnikami z oryginalnego konglomeratu (czyli $M \models \alpha,\ \alpha \in \mathcal{L}(\mathbf{S}(M)) \Rightarrow \varepsilon_{\mathbf{S}}(M) \models \alpha$). Projekcja zawęża sygnaturę konglomeratu do nowo podanej sygnatury $\mathbf{S}$. Jednakże wszystkie zależności pomiędzy konceptami, rolami i osobnikami, których nazwy pozostały w słowniku, są zachowywane (czyli $M \vDash \alpha$, $\alpha \in \mathcal{L}(\mathbf{S}) \Rightarrow \pi_{\mathbf{S}}(M) \vDash \alpha$). Formalnie definiujemy te działania jako:

$$\varepsilon_{\mathbf{S}}(M) = (\mathbf{S}(M) \cup \mathbf{S}, \{\mathcal{I} \in \mathbf{I}(\mathbf{S}(M) \cup \mathbf{S}): \mathcal{I}|\mathbf{S}(M) \in \mathbf{W}(M)\}) \tag{1}$$

$$\pi_{\mathbf{S}}(M) = (\mathbf{S}, \{\mathcal{I}|\mathbf{S}: \mathcal{I} \in \mathbf{W}(M)\}) \tag{2}$$

Działanie **przemianowania** (ρ) pozwala na zmianę nazw w sygnaturze danego konglomeratu. W tym celu wykorzystujemy pojęcie *funkcji przemianowującej* γ, będącej trójką (γ_C, γ_R, γ_I) bijekcji z $\mathcal{N}$ do $\mathcal{N}$. Przez $\gamma(\mathbf{S})$ oznaczamy sygnaturę $\gamma_C(\mathbf{C}(\mathbf{S})) \uplus \gamma_R(\mathbf{R}(\mathbf{S})) \uplus \gamma_I(\mathbf{I}(\mathbf{S}))$, a przez $\gamma(\mathcal{I})$, gdzie $\mathcal{I}$ jest $\mathbf{S}$-interpretacją, oznaczamy $\gamma(\mathbf{S})$-interpretację $\mathcal{I}'$, taką że $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ oraz $\gamma(X)^{\mathcal{I}'} = X^{\mathcal{I}}$ dla każdego $X \in \mathbf{S}$. Działanie formalnie definiujemy jako:

$$\rho_\gamma(M) = (\gamma(\mathbf{S}), \gamma(\mathbf{W})) \tag{3}$$

Operator **selekcji** (σ) działa na części interpretacyjnej konglomeratu i zawęża zbiór jego dopuszczalnych interpretacji do tych, które spełniają zadane zdanie α:

$$\sigma_\alpha(M) = (\mathbf{S}(M), \{\mathcal{I} \in \mathbf{W}(M): \mathcal{I} \vDash \alpha\}) \tag{4}$$

Unia ($\cup$), **przecięcie** ($\cap$) oraz **różnica** ($-$) konglomeratów to działania binarne, które dokonują odpowiednio sumowania, przecięcia i odjęcia zbiorów dopuszczalnych modeli konglomeratów (zakładamy, że $\mathbf{S}_1 = \mathbf{S}_2$):

$$M_1 \cup M_2 = (\mathbf{S}_1, \mathbf{W}_1 \cup \mathbf{W}_2) \tag{5}$$

$$M_1 \cap M_2 = (\mathbf{S}_1, \mathbf{W}_1 \cap \mathbf{W}_2) \tag{6}$$

$$M_1 - M_2 = (\mathbf{S}_1, \mathbf{W}_1 - \mathbf{W}_2) \tag{7}$$

Zdanie α jest spełniane przez konglomerat $M_1 \cup M_2$ wtwg. $M_1 \vDash \alpha$ **oraz** $M_2 \vDash \alpha$, zaś przez konglomerat $M_1 \cap M_2$ wtwg. $M_1 \vDash \alpha$ **lub**[3] $M_2 \vDash \alpha$. Dla operacji różnicy nie można podać prostej formuły uzależniającej spełnianie zdania α przez M' od spełniania α przez M_1 i M_2.

Operacje unii, przecięcia i różnicy wymagają, aby sygnatury obu sumowanych konglomeratów były identyczne. Wymaganie to może być jednak zniesione przez wprowadzenie operacji *uogólnionej unii konglomeratów* ($\cup_g$) zdefiniowanej jako: $M_1 \cup_g M_2 = \varepsilon_{\mathbf{S}(M_2)}(M_1) \cup \varepsilon_{\mathbf{S}(M_1)}(M_2)$. Analogicznie definiujemy działania uogólnionego przecięcia i różnicy.

Operacje różnicy i unii są *nielingwistyczne*, czyli jej zastosowanie może prowadzić do utworzenia konglomeratu, który nie odpowiada żadnemu zbiorowi zdań dialektu $\mathcal{L}$. W dalszej części artykułu problem ten będzie przedyskutowany bardziej szczegółowo.

Algebra konglomeratów wprowadza też bardziej skomplikowane operacje złączeń: **złączenia przecinającego** ($\bowtie_i$) i **złączenia sumującego** ($\bowtie_u$), formalnie zdefiniowane jako:

[3] Może tu powstać wrażenie, że spójniki „oraz" i „lub" użyte są odwrotnie. Jednak przedstawione rozumowanie jest poprawne, gdyż unia zwiększa liczbę interpretacji, a co za tym idzie, zmniejsza liczbę zdań spełnianych przez konglomerat (są to tylko zdania spełniane przez obydwa sumowane konglomeraty). Analogicznie przecięcie zmniejsza liczbę interpretacji, zwiększając tym samym liczbę spełnianych zdań.

$$M_1 \bowtie_{\mathrm{i}} M_2 = (\gamma_1(\mathbf{S}_1) \cup \gamma_2(\mathbf{S}_2), \{\mathcal{I} \in \mathbf{I}(\mathbf{S}'): \mathcal{I}|\gamma_1(\mathbf{S}_1) \in \gamma_1(\mathbf{W}_1) \wedge \mathcal{I}|\gamma_2(\mathbf{S}_2) \in \gamma_2(\mathbf{W}_2)\}) \quad (8)$$

$$M_1 \bowtie_{\mathrm{u}} M_2 = (\gamma_1(\mathbf{S}_1) \cup \gamma_2(\mathbf{S}_2), \{\mathcal{I} \in \mathbf{I}(\mathbf{S}'): \mathcal{I}|\gamma_1(\mathbf{S}_1) \in \gamma_1(\mathbf{W}_1) \vee \mathcal{I}|\gamma_2(\mathbf{S}_2) \in \gamma_2(\mathbf{W}_2)\}) \quad (9)$$

W powyższych wzorach przez $\mathbf{S}'$ oznaczono sygnaturę konglomeratu wynikowego M', czyli $\gamma_1(\mathbf{S}_1) \cup \gamma_2(\mathbf{S}_2)$. Jako operacje pomocnicze są tu wykorzystywane funkcje przemianowujące: γ_1 i γ_2. Funkcje te poprzedzają każdą nazwę konceptu i roli w sygnaturze unikalnym przedrostkiem[4]. Tym samym unika się potencjalnego konfliktu nazw pomiędzy konglomeratami. Operacje złączeń można wyrazić za pomocą innych działań. $M_1 \bowtie_{\mathrm{i}} M_2$ może być wyrażone jako $\rho_{\gamma_1}(M_1) \cap_{\mathrm{g}} \rho_{\gamma_2}(M_2)$, analogicznie $M_1 \bowtie_{\mathrm{u}} M_2 = \rho_{\gamma_1}(M_1) \cup_{\mathrm{g}} \rho_{\gamma_2}(M_2)$. To uzasadnia nazwy złączeń („przecinające" i „sumujące"), jako że możemy postrzegać je jako bezpieczny (ze względu na unikanie konfliktu nazw) sposób wykonania przecięcia i unii konglomeratów.

Ostatnią, bardzo ważną, operacją algebry konglomeratów jest **wchłonięcie** (υ_C). Wchłonięcie odnosi do siebie dziedziny interpretacji konglomeratów będących argumentami działania ($C \in \mathcal{L}_{\mathrm{C}}(\mathbf{S}_2)$):

$$M_1 \upsilon_C M_2 = (\mathbf{S}_1 \cup \mathbf{S}_2, \{\mathcal{I} \in \mathbf{I}(\mathbf{S}_1 \cup \mathbf{S}_2): \mathcal{I}|\mathbf{S}_2 \in \mathbf{W}_2 \wedge (\mathcal{I}|\mathbf{S}_1 \cap C^{\mathcal{I}}) \in \mathbf{W}_1\}) \quad (10)$$

W definicji działania użyto *zawężenia* **S**-*interpretacji* $\mathcal{I}$: pisząc $\mathcal{I} \cap \Delta'$, mamy na myśli interpretację $\mathcal{I}' = (\Delta', \cdot^{\mathcal{I}'})$, taką że $X^{\mathcal{I}'} = X^{\mathcal{I}} \cap \Delta'$ dla każdego $X \in \mathbf{S}$. Innymi słowy, zawężenie interpretacji ogranicza jej dziedzinę do zadanego zbioru Δ'. Jak widać z definicji operacji wchłonięcia, każdy model wynikowego konglomeratu M' musi mieć odpowiadający model z M_2 (taki, który opisuje zależności spełnione w całej dziedzinie) oraz odpowiadający model z M_1, ale taki, który opisuje zależności spełnione w części dziedziny odpowiadającej zakresowi danego konceptu C. Mówiąc może trochę bardziej obrazowo, operacja wchłonięcia powoduje, że zależności między konceptami, rolami i osobnikami z modułu M_1 nadal obowiązują, ale tylko we fragmencie dziedziny nowych modeli.

3.3. Semantyczne cechy konglomeratów

W tym podrozdziale zawarto uwagi dotyczące specyficznych cech algebry konglomeratów, które wiążą się z zastosowanym podejściem semantycznym. Przedyskutujemy tu problem *nielingwistyczności* konglomeratów. Dla ułatwienia dyskusji tego problemu sformalizujemy najpierw pojęcie *konglomeratu lingwistycznego*.

Definicja 2 (konglomerat lingwistyczny). Konglomerat $M = (\mathbf{S}, \mathbf{W})$ nazywamy *lingwistycznym* wtwg. istnieje zbiór zdań $Z \subseteq \mathcal{L}(\mathbf{S})$, taki że $\mathbf{W} = \{\mathcal{I} \in \mathbf{I}(\mathbf{S}): \forall\alpha \in Z: \mathcal{I} \vDash \alpha\}$.

Innymi słowy konglomerat lingwistyczny to konglomerat, którego modele stanowią zbiór wszystkich modeli pewnego zbioru zdań Z. Konglomeraty lingwistyczne będziemy też ozna-

[4] Taki zabieg jest często stosowany w formalizmach opisujących ontologie modularne, a poprzedzone przedrostkiem nazwy konceptów (A, B) i ról atomowych (R, S) oznacza się jako 1:A, 2:B, 1:R itp.

czali jako $M(\mathbf{S}, Z)$, gdzie Z jest zbiorem zdań z definicji 2, a $\mathbf{S}$ to sygnatura konglomeratu. Notację tę możemy uprościć do $M(Z)$, jeśli sygnaturę można odgadnąć z kontekstu, np. $M(\{A \sqsubseteq B, A(a)\})$ jest konglomeratem o sygnaturze $\{A, B, a\}$.

W ogólności, kiedy używamy niektórych działań algebry konglomeratów, nie mamy gwarancji, że ich wynikiem będzie konglomerat lingwistyczny, nawet jeśli jego wszystkie argumenty działania to konglomeraty lingwistyczne. Działania te nazywane są *działaniami nielingwistycznymi*, a należą do nich: unia ($\cup$), różnica (–), projekcja (π) oraz (pośrednio, ze względu na nielingwistyczność unii) złączenie sumujące ($\bowtie_u$). Użycie tych operacji może spowodować wyprodukowanie konglomeratu M, który nie odpowiada żadnemu zbiorowi zdań z $\mathcal{L}(\mathbf{S}(M))$.

Przykład 1 (konglomerat nielingwistyczny). Rozważmy unię M' dwóch konglomeratów lingwistycznych $M' = M(\{A \sqsubseteq B\}) \cup M(\{B \sqsubseteq A\})$. Zauważmy, że nie jest prawdą, że $M' \models A \sqsubseteq B$ ani $M' \models B \sqsubseteq A$. Jednakże jeśli wykonamy działanie: $M'' = M' \cap_g M(\{A \sqcap \neg B(a), \neg A \sqcap B(b)\})$, w jego wyniku otrzymamy konglomerat o pustym zbiorze modeli, $\mathbf{W}(M'') = \varnothing$. Choć zatem konglomerat M' nie spełniał żadnego ze zdań $A \sqsubseteq B$, $B \sqsubseteq A$, to wprowadzenie do niego asercji przeczących obydwu tym zdaniom spowodowało sprzeczność.

Istnienie konglomeratów nielingwistycznych może wydawać się problemem z punktu widzenia omawianej algebry. Jednakże, według autorów artykułu, możliwość operowania na konglomeratach nielingwistycznych stanowi zaletę przedstawianej propozycji. Działania nielingwistyczne pozwalają użytkownikowi na zwiększenie liczby dozwolonych interpretacji (oferując tym samym możliwości podobne do oferowanych przez niektóre silniki wnioskujące funkcji wycofywania – ang. *retract* – zdań z bazy wiedzy), a także na przechowywanie kilku alternatywnych zbiorów aksjomatów i asercji w jednym konglomeracie. Możliwości te zostaną zilustrowane w następnym rozdziale.

4. Przykłady użycia algebry konglomeratów

Podobnie jak algebra relacyjna, algebra konglomeratów może być postrzegana jako podstawa pewnego języka, którym może posługiwać się końcowy użytkownik bazy wiedzy. Taki język posiadałby pożądaną właściwość domknięcia: byłby używany do przetwarzania konglomeratów (ekstrakcji ich fragmentów, łączenia ich itp.), a w wyniku otrzymywalibyśmy nowy konglomerat, lepiej dostosowany do zaspokojenia konkretnej potrzeby użytkownika.

Poniższy przykład pokazuje, w jaki sposób użytkownik mógłby połączyć informacje z dwóch konglomeratów, aby uzyskać pożądaną postać wiedzy, na której mu zależy (we wszystkich przykładach w tym rozdziale przyjęto założenie, że wykorzystywany jest dialekt $\mathcal{L}$ o wystarczającej ekspresywności).

Przykład 2 (przecięcie konglomeratów). Rozważmy dwa konglomeraty: M_1 opisuje zagadnienia kadrowe, a M_2 strukturę organizacyjną szpitala:

$$M_1 = M(\{\exists zarządza.Jednostka \sqsubseteq Specjalista, Specjalista \sqsubseteq Pracownik\})$$

$$M_2 = M(\{jestOrdynatorem(janKowalski, neurochirurgia), Oddział(neurochirurgia)\})$$

Chcemy połączyć informacje z dwóch konglomeratów i wywnioskować, że *janKowalski* jest *Specjalistą*. W tym celu tworzymy przecięcie konglomeratów $M' = M_1 \cap_g M_2$, a następnie zawężamy zbiór jego modeli przez wprowadzenie dwóch dodatkowych „pomostowych"[5] aksjomatów wiążących znaczenie terminów z M_1 i M_2: $M'' = M' \cap_g M(\{jestOrdynatorem \sqsubseteq zarządza, Oddział \sqsubseteq Jednostka\})$. Ten ciąg działań (ostatni krok można przeprowadzić również za pomocą dwukrotnego wykonania operacji selekcji) pozwala osiągnąć wyznaczony cel, jako że $M'' \vDash Specjalista(janKowalski)$.

W powyższym przykładzie nie wystąpił konflikt nazw pomiędzy łączonymi konglomeratami. W ogólności konflikt taki może występować, a w takich przypadkach pomocne może okazać się wykorzystanie złączeń. Poniższy przykład pokazuje, jak można połączyć informacje z dwóch konglomeratów, gdzie te same terminy zostały wykorzystane do wyrażenia innych znaczeń.

Przykład 3 (złączenie konglomeratów). Ponownie rozważmy dwa konglomeraty: M_1 i M_2. Oba konglomeraty zawierają informacje o lokalizacjach do wynajęcia i maję tę samą sygnaturę $\mathbf{S} = \{LNS, LŚS, LWS\}$, gdzie poszczególne koncepty opisują odpowiednio lokalizacje o niskim, średnim i wysokim standardzie. W obu konglomeratach koncepty *LNS*, *LŚS* i *LWS* są parami rozłączne i pokrywają swym zakresem całą dziedzinę zainteresowań. Jednakże w M_1 i M_2 zostały użyte inne kryteria oceny standardu lokalizacji. W pierwszym przypadku szukano kempingu do zatrzymania się na jedną noc, a w drugim miejsca do wynajęcia na okres miesiąca lub dłuższy. Naszym celem jest import informacji zawartej w M_1 do M_2, tak by ponownie wykorzystać raz przeprowadzoną ocenę:

1. W pierwszym kroku dokonamy złączenia przecinającego konglomeratów. W wyniku otrzymamy konglomerat $M' = M_1 \bowtie_i M_2$. Koncepty zostały przemianowane, tak że $\mathbf{S}(M') = \{1{:}LNS, 2{:}LNS, 1{:}LŚS, 2{:}LŚS, 1{:}LWS, 2{:}LWS\}$.
2. Następnie ujawniamy w ontologii znane nam kryteria oceny. W tym przykładzie użyjemy tylko jednego kryterium: posiadanie łazienki. Rozszerzamy zatem sygnaturę M', budując konglomerat $M'' = \varepsilon_{\{LokumZŁazienką\}}(M')$.
3. W kolejnym kroku związujemy kryteria z oceną. Założymy, że wiemy, że w konglomeracie M_1 wszystkie lokalizacje z łazienką były uznawane za lokalizacje o wysokim standardzie (w M_1 ocenialiśmy standard lokalizacji z punktu widzenia

[5] Nawiązujemy tutaj do *reguł pomostowych* definiowanych w celu ustanowienia zależności pomiędzy terminami z dwóch różnych ontologii [3].

jednego noclegu). Z kolei według kryteriów stosowanych dla M_2 (znacznie dłuższy pobyt), żadna lokalizacja bez łazienki nie może być uznana za lokalizację o standardzie innym niż niski. Wprowadzamy te informacje do konglomeratu za pomocą następującego działania:

4. $M''' = M'' \cap_g M(\{LokumZŁazienką \sqsubseteq 1{:}LWS, \neg LokumZŁazienką \sqsubseteq 2{:}LNS\})$
5. Naturalnie drugi aksjomat jest poprawny tylko wtedy, gdy zakładamy, że cała nasza dziedzina zainteresowań to lokalizacje (wyżej przyjęliśmy takie założenie).
6. Na koniec usuwamy niechciane terminy z sygnatury konglomeratu: $M = \pi_{\{2:LNS,\ 2:LŚS,\ 2:LWS\}}(M''')$.

W powyższych krokach dokonaliśmy tłumaczenia znaczenia terminów zawartych w jednym konglomeracie (module) na znaczenie wykorzystywane w innym (takie, które chcieliśmy zachować). Warto zauważyć, że nasza założona wiedza o kryteriach oceny pozwalała na wykonanie właśnie takiego tłumaczenia — innymi słowy wykorzystaliśmy maksimum dostępnych nam informacji. Dzięki przeprowadzonym krokom wszystkie lokalizacje niskiego i średniego standardu według słownictwa M1 zostały uznane za lokalizacje niskiego standardu (z powodu braku łazienki) według słownictwa M2.

Co ciekawie, przedstawiony w przykładzie 5 zestaw kroków odpowiada zarysowanemu w pracy [4] procesowi łączenia informacji z dwóch kontekstów. Podkreśla to, że algebra konglomeratów jest dobrze umocowana w kontekstowym podejściu do modelowania wiedzy.

Zauważmy także, że w obu przykładach (przykłady 4 i 5) wykorzystano wyłącznie zdania logiki opisowej do odwzorowania znaczenia terminów z dwóch różnych konglomeratów. Nie użyto tutaj zatem żadnego dodatkowego języka, tak jak proponowane jest to w metodach modularyzacji, takich jak DDL [3] czy ε-connections [5].

Może wydawać się, że istnienie działań nielingwistycznych narusza powyższe założenie. Jednakże, jak pokazano w dalszej części tekstu, nie należy uważać tych działań za czynnik rozszerzający ekspresywność konglomeratów. Operacje nielingwistyczne pozwalają na zwięzłe ujęcie działań, do wykonania których musielibyśmy wykorzystać kilka modułów. Można zatem traktować je jako zwięzły sposób opisania swego rodzaju „operacji wsadowych" na zbiorach konglomeratów. Poniższy przykład ilustruje i uzasadnia to stwierdzenie.

Przykład 4 (unia konglomeratów). Rozważmy konglomerat M:

$$\begin{aligned} M = M(\{ & \top \sqsubseteq\ \leq 1\ zamordował.\{ofiara\}, \\ & \exists oskarża^{-}.GodnyZaufaniaŚwiadek \sqsubseteq \exists zamordował.\{ofiara\}, \\ & GodnyZaufaniaŚwiadek \sqsubseteq \exists byłObecny.MiejsceZbrodni\}) \end{aligned}$$

Powyższy moduł zawiera wiedzę terminologiczną mówiącą, że istnieje tylko jeden morderca, a jest nim ten, kto jest oskarżony przez godnego zaufania świadka. Godny zaufania świadek musi być obecny na miejscu zbrodni.

Chcielibyśmy rozważyć dwie (wzajemnie wykluczające się) wersje wydarzeń:

$$M_1 = M \cap_g M(\{\ \mathit{GodnyZaufaniaŚwiadek}(\mathit{janSzemrany}),\ \mathit{oskarża}(\mathit{janSzemrany}, \mathit{piotrNiewinny})\ \})$$

$$M_2 = M \cap_g M(\{\ \mathit{GodnyZaufaniaŚwiadek}(\mathit{henrykNieposzlakowany}),\ \mathit{oskarża}(\mathit{henrykNieposzlakowany}, \mathit{marekWinny})\ \})$$

Te dwie wersje wydarzeń mogą być przechowywane w jednym konglomeracie $M' = M_1 \cup_g M_2$. Żaden wniosek co do winy poszczególnych osobników nie może być jednoznacznie wyciągnięty z M'. Dodanie kolejnej informacji do konglomeratu M', informacji, że *janSzemrany* nie był obecny na miejscu zbrodni, pozwala na wykluczenie jednej z wersji i określenie mordercy: $M'' = M' \cap M(\{\neg\exists \mathit{byłObecny}.\mathit{MiejsceZbrodni}(\mathit{janSzemrany})\})$; wówczas już znamy mordercę, gdyż $M'' \vDash \mathit{zamordował}(\mathit{marekWinny}, \mathit{ofiara})$.

Powyższy przykład ilustruje, jak za pomocą unii można wykonać operacje, do których przeprowadzenia trzeba byłoby wykorzystać dwa konglomeraty.

Inny przykład wykorzystania algebry konglomeratów to optymalizacja wykonania zapytań. W trakcie prac nad systemem RKaSeA opracowano pierwszą wersję algorytmu, który pozwala na automatyczną dekompozycję złożonej bazy wiedzy na zbiór konglomeratów. Konglomeraty można ponownie złożyć do postaci całej bazy wiedzy lub jej fragmentu za pomocą działań algebry konglomeratów, przy czym znane i pamiętane jest wyrażenie, które może służyć do takiego odtworzenia. Podczas udzielania odpowiedzi zapytanie użytkownika jest analizowane i podejmowana jest decyzja, które konglomeraty muszą być wykorzystane. Skutkiem tego zadanie wskazane przez użytkownika wykonywane jest na fragmencie bazy wiedzy, potencjalnie znacznie mniejszym niż cała baza, co wpływa na znaczne zwiększenie wydajności silnika wnioskującego, a w rezultacie pozwala także na obsłużenie baz wiedzy o większej ekspresywności.

5. Konglomeraty i rozstrzygalność

Ze względu na fakt, że użycie algebry konglomeratów może doprowadzić do wykroczenia poza oryginalną ekspresywność stosowanego dialektu logiki opisowej, należy zadać pytanie, czy podstawowe problemy wnioskowania pozostają dla konglomeratów rozstrzygalne.

W rozdziale tym skupiamy się na problemie spełnialności konceptów (*c-sat*), który dla konglomeratów może być sformułowany następująco: dla danego konglomeratu M i konceptu $C \in \mathcal{L}_C(\mathbf{S}(M))$ sprawdź, czy istnieje interpretacja $\mathcal{I} \in \mathbf{W}(M)$, dla której $C^{\mathcal{I}} \neq \emptyset$.

Czynnikami, które wpływają na rozstrzygalność problemu, są: zastosowany dialekt logiki opisowej $\mathcal{L}$ oraz zbiór wykorzystywanych działań algebry konglomeratów. Każdy problem może zatem być opisany trójką (p, $\mathcal{L}$, A), gdzie p to nazwa problemu, $\mathcal{L}$ to zastosowany dialekt, a A to wykorzystywany zbiór prymitywnych działań algebry konglomeratów (przez **A** oznaczamy zbiór wszystkich działań: $\{\varepsilon,\pi,\rho,\sigma,\cup,\cap,-,\upsilon\}$). Co więcej, zakładamy, że jako stałe wykorzystać możemy wyłącznie konglomeraty lingwistyczne, zatem problem rozstrzygalności ograniczamy tylko do konglomeratów możliwych do skonstruowania z konglomeratów lingwistycznych za pomocą wybranego zbioru działań algebry. Wstępne rezultaty analizy zawarte są w poniższych twierdzeniach.

Twierdzenie 1. Problem (*c-sat*, $\mathcal{SHOIQ}$, **A** – {–}) jest rozstrzygalny.

Twierdzenie 2. Problem (*c-sat*, $\mathcal{ALB}$, **A**) jest rozstrzygalny.

Twierdzenia te, ze względu na konieczność oszczędności miejsca, przytaczamy bez dowodu. Szkicowo dowód ten przebiega przez przypisanie konglomeratom nie zbiorów zdań, a zbiorów zbiorów zdań (przy czym konglomeratom lingwistycznym $M(Z)$ przypisywany jest po prostu zbiór $\{Z\}$). Sprawdzenie spełnialności konceptu C polega na sprawdzeniu jego spełnialności względem wszystkich zbiorów zdań przypisanych danemu konglomeratowi za pomocą standardowej procedury wnioskującej dla dialektu $\mathcal{SHOIQ}$ [9] i $\mathcal{ALB}$ [10]. Trudność z działaniem różnicy polega na tym, że wymaga ona negowania zdań z przypisanych zbiorów, w tym asercji typu $R(a, b)$, co wymaga istnienia w $\mathcal{L}$ negacji roli.

Dwa powyższe twierdzenia pokazują, że zakres zastosowania algebry konglomeratów jest stosunkowo szeroki. Naturalnie pojawia się pytanie, czy rezultat twierdzenia 2 można rozszerzyć na bardziej ekspresywne logiki opisowe. Kwestia ta jest wciąż otwarta, choć raczej należy wykazywać tu pesymizm, gdyż argumentacja zastosowana w dowodzie twierdzenia 2 wymaga negowania roli, a rozszerzenie bardziej ekspresywnych dialektów o negację roli prowadzi do nierozstrzygalności.

6. Prace o podobnej tematyce

Algebra konglomeratów inspirowana jest algebrą relacji oryginalnie zaproponowaną przez Codda [2]. W naszych pracach oparliśmy się jednak raczej na wersji zaprezentowanej w [11], która różni się od oryginału wprowadzeniem formalizmu do obsługi nazw kolumn (u Codda były one rozróżniane wyłącznie przez kolejność).

Nieco podobne podejście do zagadnienia można zaobserwować w zaproponowanej ostatnio ([12]) algebrze modułów oprogramowania. Każdy moduł charakteryzowany jest tam przez zbiór dopuszczalnych (zgodnych z jego specyfikacją) implementacji. Operacje algebraiczne na modułach są operacjami na zbiorach dopuszczalnych implementacji. Praca [12] prezentuje jednak inne motywacje – autorzy poszukują tam pojedynczej operacji algebraicznej, która w jak najlepszy sposób opisze kompozycję modułów.

W dziedzinie modularyzacji ontologii istnieje wiele propozycji metod i technik: najbardziej znaczące to DDL [3], ε-connections [5] oraz P-DL [13] (przegląd metod zainteresowany Czytelnik znajdzie w [14]). Każda z tych metod definiuje jednak stosunkowo wąski zakres celów. DDL i ε-connections skupiają się na zagadnieniu połączenia istniejących ontologii za pomocą dodatkowych języków: reguł pomostowych lub dodatkowych specjalnych ról (relacji). P-DL z kolei wprowadza strukturę zagnieżdżonych modułów różniących się stopniem zakapsułkowania (sposobem udostępniania wiedzy innym modułom). W centrum zainteresowania tych metod jest przepływ wniosków pomiędzy złączonymi ontologiami, a co za tym idzie zagadnienia w rodzaju lokalności (czy wykorzystanie wiedzy z ontologii O_1 przez ontologię O_2 nie „uszkodzi" tej pierwszej, np. nie spowoduje jej sprzeczności). Tymczasem w algebrze konglomeratów przyjmowane jest założenie o konieczności udostępnienia możliwości budowy nowego konglomeratu dostosowanego do potrzeb użytkownika (choć problem *lokalności* [6] może być zapisany jako $\pi_{\mathbf{S}}(M') = M$, gdzie M jest modułem importowanym, **S** jego sygnaturą, a M' modułem importującym, a sama algebra konglomeratów może stanowić dobry punkt wyjścia do opisu zależności między modułami i ontologiami).

W kilku pracach związanych z przetwarzaniem baz wiedzy i ontologii pojawia się pojęcie algebry. Jeden z przykładów to algebra ontologii prezentowana w [15]. Autorzy [15] kierują się podobnymi motywacjami, wskazując, że twórcy ontologii zawsze przyjmują pewien kontekst ich wykorzystania. W [15] wykorzystano jednak podejście ściśle syntaktyczne: ontologie przedstawiane są tu jako grafy zbliżone strukturą do grafów dokumentu RDF. Wykonanie jakiejkolwiek operacji binarnej na dwóch ontologiach wymaga zdefiniowania dla nich tzw. *reguł artykulacji*. Reguły artykulacji określają, w jaki sposób stworzyć krawędzie grafu połączonych ontologii. Algebra proponowana w [15] definiuje operacje sumy, różnicy i przecięcia ontologii jako działania dające w wyniku grafy zawierające pewne podzbiory połączonego zbioru zdań utworzonego w wyniku zastosowania reguł artykulacji.

Podejście syntaktyczne zastosowano we wstępnej propozycji algebry modułów ontologicznych zaproponowanej w ramach projektu NeOn [16]. Algebra ta charakteryzuje się podobieństwem do przedstawionej w niniejszym artykule propozycji: zdefiniowane są tam działania sumy, różnicy i przecięcia modułów, dodatkowo każdy moduł wyposażony jest w interfejs, czyli zbiór widocznych na zewnątrz nazw konceptów, ról i osobników. W algebrze modułów projektu NeOn brak jednak złączeń, a wszystkie działania są w rzeczywistości operacjami na zbiorach zdań. Podejście takie może rodzić trudności, które najlepiej chyba ilustruje definicja różnicy modułów z [16]: różnica $(M_1 - M_2)$ określona jest tu jako moduł (zbiór zdań), taki że, dla dowolnego zdania α, α wynika z $(M_1 - M_2)$ wtwg. α wynika z M_1 oraz nie wynika z M_2. Problem z przytoczoną definicją stanie się jasny, jeśli za α podstawimy jakąkolwiek tautologię.

7. Podsumowanie

W niniejszym artykule zaprezentowaliśmy algebrę konglomeratów. Pojęcie konglomeratu jest ściśle semantyczne, a każdy konglomerat jest zbiorem swoich modeli.

Algebra konglomeratów dostarcza szerokiej gamy narzędzi do manipulacji wiedzą. Stanowi także dobrą podstawę do opisu wielu zagadnień związanych z modularyzacją baz wiedzy (jak np. optymalizacja zapytań). Dostarcza również ich użytkownikom metod przekształcania konglomeratów do interesującej ich postaci, tak że mogą oni dostosowywać sposób modularyzacji ontologii do swoich bieżących potrzeb.

Algebra konglomeratów została stworzona na podstawie algebry relacji. Bardzo ciekawe kierunki jej rozwoju wiążą się z możliwością wprowadzenia do niej analogów rozszerzeń algebry relacji: zapytań typu *for* i *while* [17] oraz elementów algebr relacji zagnieżdżonych [18].

Oczywiście, dalszy rozwój algebry konglomeratów związany jest przede wszystkim z jej praktycznymi zastosowaniami: implementacją systemu RKaSeA zawierającą algebraiczny mechanizm dekompozycji bazy wiedzy i optymalizacji zapytań, a także obsługującą język zapytań oparty na algebrze konglomeratów.

BIBLIOGRAFIA

1. Baader F. A., McGuiness D. L., Nardi D., Patel-Schneider P. F. (red.): The Description Logic Handbook: Theory, implementation, and applications, Cambridge University Press, 2003.
2. Codd, E. F.: Relational Completeness of Data Base Sublanguages. Database Systems, 1979, Vol. 6, s. 65÷98.
3. Serafini L., Tamilin A. Reasoning with Instances in Distributed Description Logics. Raport badawczy, Fondazione Bruno Kessler – IRST, 2007. Dostępny pod adresem: http://sra.itc.it/people/tamilin/publications/2007/iswc/tr.pdf.
4. Benerecetti, M., Bouquet, P., Ghidini, C.: Contextual reasoning distilled. Journal of Experimental & Theoretical Artificial Intelligence, 2000, Vol. 12, No. 3, s. 279÷305.
5. Cuenca-Grau B., Parsia B., Sirin E.: Tableau Algorithms for $\mathcal{E}$-Connections of Description Logics. Raport badawczy UMIACS, Maryland, 2004.

6. Grau B. C., Horrocks I., Kazakov Y., Sattler U.: A logical framework for modularity of ontologies. Proceedings of the 20th International Joint Conference on Artificial Intelligence, s. 298÷303, IJCAI, 2007.
7. Ghidini C., Serafini L., Tessaris S.: On relating heterogeneous elements from different ontologies. Proceedings of the 20th International Workshop of Description Logics, Bozen-Bolzano 2007, s. 283÷290.
8. Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with Rules. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 2005, Vol. 3, No. 1, s. 41÷60.
9. Horrocks, I., Sattler, U.: A Tableau Decision Procedure for SHOIQ. Journal of Automated Reasoning, 2007, Vol. 39, No. 3, s. 249÷276.
10. Tobies, S.: Complexity Results and Practical Algorithms for Logics in Knowledge Representation. Praca doktorska, RWTH Aachen 2001.
11. Hall, P., Hitchcock, P., Todd, S.: An algebra of relations for machine computation. Proceedings of the 2nd ACM SIGACT-SIGPLAN symposium on Principles of programming languages, ACM, 1975, s. 225÷232.
12. Herrmann C., Krahn H., Rumpe B., Schindler B., Völkel S.: An Algebraic View on the Semantics of Model Composition. Model Driven Architecture – Foundations and Applications, Springer, 2007, s. 99÷113.
13. Bao J., Caragea D., Honavar V. G.: Modular Ontologies - A Formal Investigation of Semantics and Expressivity. Proceedings of the First Asian Semantic Web Conference, Springer-Verlag, 2006, s. 616÷631.
14. Goczyła K., Waloszek A., Waloszek W.: Techniki modularyzacji ontologii. Bazy danych. Rozwój metod i technologii – Architektura, metody formalne i zaawansowana analiza danych, red.: S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek, WKŁ, 2008, s. 309÷322,.
15. Mitra P., Wiederhold G.: An Ontology-Composition Algebra. Handbook on Ontologies, Springer-Verlag, 2004, s. 171÷216.
16. d'Aquin M. i in.: NeOn Formalisms for Modularization: Syntax, Semantics, Algebra. Raport D1.1.3 projektu NeOn, http://www.neon-project.org, 2008.
17. Aho, A. V., Ullman, J. D.: Universality of data retrieval languages. Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, ACM, 1979, s. 110÷119.
18. Colby, L. S.: A recursive algebra and query optimization for nested relations. SIGMOD Rec., 1989, Vol. 18, No. 2, s. 273÷283.

Recenzent: Dr inż. Michał Świderski

Wpłynęło do Redakcji 3 lutego 2009 r.

Abstract

Modularity has been recently recognized as a key requirement for collaborative ontology engineering and distributed ontology reuse. This article presents a novel approach to modularization of ontologies which is based on introduced notion of conglomeration algebra. A conglomeration is a semantic form of an ontology module. The conglomeration algebra defines several operations on conglomerations. The set of operations is inspired by relational algebra introduced by Codd and embraces extension (ε), selection (σ), rename (ρ), projection (π), union (∪), intersection (∩), difference (–), joins (⋈), and absorption (υ). At the foundation of the formalism lies the conviction that the way of modularization introduced by the authors of the ontology must sometimes be overridden by specific user needs. Following this idea, the algebra puts a stress on delivering to the user various methods of manipulation of conglomerations.

The article presents typical scenarios of use of the algebra. Algebra can be used for providing the end-user of a knowledge base with methods of merging and aligning two or more conglomerations. Specific features of conglomeration algebra allow for using it to model information retraction and more unconventional use of modularity, like holding several alternative versions of knowledge in a single conglomeration. A scenario of use of the algebra for optimization of the query-answering process has also been presented.

In our paper we discuss the issue of decidability of basic inference problems for conglomerations and present two theorems, which suggest that the range of use of s-module algebra is fairly broad.

We conclude the paper with detailed comparison between the proposed algebra and related work. In the summary we also sketch the directions of further development of the presented methods.

Adresy

Krzysztof GOCZYŁA: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. G. Narutowicza 11/12, 80-952 Gdańsk, Polska, kris@eti.pg.gda.pl.

Aleksander WALOSZEK: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. G. Narutowicza 11/12, 80-952 Gdańsk, Polska, alwal@eti.pg.gda.pl.

Wojciech WALOSZEK: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. G. Narutowicza 11/12, 80-952 Gdańsk, Polska, wowal@eti.pg.gda.pl.