



POLITECHNIKA GDAŃSKA
Wydział Elektroniki, Telekomunikacji
i Informatyki



Aleksander Waloszek

Hierarchiczna kontekstualizacja baz wiedzy

Rozprawa doktorska

Promotor:

dr hab. inż Krzysztof Goczyła
prof. nadzw. Politechniki Gdańskiej
Wydział Elektroniki, Telekomunikacji
i Informatyki
Politechnika Gdańska

Gdańsk, 2010

1	Wstęp.....	1
2	Ontologie w zastosowaniach informatycznych.....	7
2.1	Ontologie klasyczne	7
2.1.1	Filozoficzne pojęcie ontologii.....	7
2.1.2	Logika pierwszego rzędu jako język opisu wiedzy.....	8
2.1.3	Rozwinięcia logiki pierwszego rzędu	13
2.1.4	Formalizacja pojęcia ontologii	17
2.2	Ontologie w Internecie	19
2.2.1	Założenia inicjatywy Semantic Web.....	20
2.2.2	Logika opisowa	20
2.2.3	Systemy wnioskujące i języki	25
2.3	Podsumowanie.....	27
3	Normalizacja ontologii	29
3.1	Metoda walidacji ontologii OntoClean.....	29
3.1.1	Metawłaściwości w procesie konceptualizacji.....	30
3.1.2	Ograniczenia wynikające z metawłaściwości	33
3.1.3	Dodatkowe metawłaściwości	35
3.1.4	Przykłady.....	35
3.1.5	Metoda OntoClean a projektowanie relacyjnej bazy danych.....	38
3.2	Zastosowanie ontologii fundacjonistycznych.....	39
3.3	Podsumowanie.....	40
4	Modularyzacja ontologii	43
4.1	Konteksty w logice	43
4.1.1	Wprowadzenie pojęcia kontekstu do logiki	45
4.1.2	Logika zaimków wskazujących	49
4.1.3	Kontekstowa logika zdań (PLC)	52
4.1.4	Semantyka modeli lokalnych/systemy wielokontekstowe (LMS/MCS)	59
4.2	Modularyzacja ontologii w Semantic Web	64
4.2.1	Rozproszona logika opisowa (DDL).....	65
4.2.2	\mathcal{E} -Connections	69
4.3	Podsumowanie.....	73
5	Metoda SIM hierarchicznej kontekstualizacji ontologii	75
5.1	Pojęcia związane z kontekstualizacją hierarchiczną.....	75
5.2	Semantyka ontologii hierarchicznej	80
5.3	Wnioskowanie w kontekstowej bazie wiedzy	82
5.3.1	Problemy wnioskowania kontekstowego	82
5.3.2	Wnioskowanie integracyjne	84
5.3.3	Wnioskowanie strukturalne.....	93
5.4	Podsumowanie.....	100
6	Projektowanie hierarchicznych baz wiedzy	101
6.1	Cechy systemu zgodnego z SIM	101
6.1.1	Struktura bazy wiedzy a parametry kontekstowe.....	102
6.1.2	Schemat i instancje bazy wiedzy.....	108
6.1.3	Opis działania systemu zgodnego z SIM	110
6.1.4	Metoda OntoClean a hierarchiczna kontekstualizacja bazy wiedzy	113
6.2	Problemy modelowania	119
6.2.1	Modelowanie różnych punktów widzenia.....	120
6.2.2	Konteksty jako obiekty.....	123
6.3	Podsumowanie.....	131
7	Praktyczna realizacja ontologii zgodnej z metodą SIM	132

7.1	System KaSeA	132
7.2	Ontologia PIPSDrugs	134
7.2.1	Charakterystyka ontologii PIPSDrugs	135
7.2.2	Kontekstualizacja ontologii PIPSDrugs	137
7.2.3	Uzyskane wyniki	139
7.3	Podsumowanie.....	141
8	Kierunki rozwoju metody SIM	142
9	Podsumowanie	145
	Bibliografia.....	150
	Wykaz definicji	158

Wykaz ważniejszych oznaczeń

$A \times B$	– iloczyn kartezjański zbiorów A i B
$A \setminus B$	– różnica zbiorów A i B
$\mathcal{P}(A)$	– zbiór potęgowy zbioru A
A_B	– zbiór wszystkich funkcji z A do B
$A \subseteq B$	– zawieranie się zbioru A w zbiorze B
$A \uplus B$	– suma rozłączna zbiorów A i B
$\alpha, \beta, \gamma \dots$	– formuły atomowe w logice
$\varphi, \chi, \psi \dots$	– formuły (atomowe i złożone) w logice
p, q, r	– zdania w logice zdaniowej
\forall, \exists	– kwantyfikator ogólny i egzystencjalny
$S \models \varphi$	– formuła φ wynika ze zbioru zdań S
$S \vdash \varphi$	– formułę φ da się wyprowadzić ze zbioru zdań S
$S \not\models \varphi$	– formuła φ nie wynika ze zbioru zdań S
$\mathcal{I} \models \varphi$	– interpretacja \mathcal{I} spełnia formułę φ
$\mathcal{I} \not\models \varphi$	– interpretacja \mathcal{I} nie spełnia formuły φ
\square, \diamond	– operatory konieczności i możliwości w logice modalnej
k, k_1, k_n	– konteksty, moduły ontologiczne
$1, 0$	– prawda, fałsz
$\text{dom}(R)$	– dziedzina relacji R
$\text{rng}(R)$	– przeciwdziedzina relacji R
$\text{Fnc}(f)$	– f jest funkcją
(a, b, c)	– ciąg
$(a, b).(c, d)$	– konkatencja ciągów
\mathbb{N}	– zbiór liczb naturalnych
wtwg	– wtedy i tylko wtedy, gdy
$\ A\ $	– liczebność zbioru A
R^*	– dopełnienie tranzytywno-zwrotne relacji R
$[a, b]$	– przedział obustronnie domknięty
$\langle a, b \rangle$	– przedział obustronnie otwarty
$[a, b), \langle a, b]$	– przedział częściowo otwarty
A, B, G, H	– koncepty atomowe w logice opisowej
C, D, E	– koncepty złożone w logice opisowej
P, Q, R	– role atomowe w logice opisowej
$C \sqcap D$	– przecięcie konceptów złożonych w logice opisowej
$C \sqcup D$	– suma konceptów złożonych w logice opisowej
$C \sqsubseteq D$	– subsumcja konceptów złożonych w logice opisowej
$\prod_{A \in \text{Sig}(O)}$	– przecięcie wszystkich konceptów atomowych występujących w sygnaturze ontologii O
$R_1 \circ R_2$	– łańcuch ról w logice opisowej

Konwencje typograficzne

<i>ontologia</i>	– nowo wprowadzane terminy oraz cytaty w oryginalnym brzmieniu
<i>blocks</i>	– nazwy funkcji pozalogicznych
<i>Above</i>	– nazwy predykatów pozalogicznych
<i>ist</i>	– nazwy predykatów logicznych

1 Wstęp

Prace, których wyniki są prezentowane w niniejszej rozprawie doktorskiej, plasują się w dziale badań nad sztuczną inteligencją nazywanym reprezentacją wiedzy i wnioskowaniem (ang. *Knowledge Representation and Reasoning*, w skrócie *KRR*). Dział ten zajmuje się sposobami przedstawiania wiedzy oraz metodami automatycznego wnioskowania. W ramach tego działu ważną rolę odgrywa nurt związany z reprezentowaniem wiedzy w sposób symboliczny, a w jego ramach poczesne miejsce zajmują formalizmy logiczne. Ma to uzasadnienie w historycznym rozwoju nauki, w trakcie którego, począwszy od starożytnych filozofów, starano się przedstawiać różnego rodzaju teorie w jak najbardziej ścisły sposób. Miało to na celu precyzyjne przekazanie myśli i uniknięcie, na ile to możliwe, niewłaściwych interpretacji. Pierwszym ze znanych formalizmów logicznych był rachunek zdań kategorycznych Arystotelesa. Współczesne formalizmy logiczne zaczęły powstawać dopiero po epoce Oświecenia, a ich rozkwit nastąpił w XX wieku. W jego drugiej połowie rozpoczęła się era informatyki, której źródła należy doszukiwać się w dążeniach do automatyzacji procesów wnioskowania, a która z drugiej strony spowodowała zintensyfikowanie prac nad różnego rodzaju formalizmami pozwalającymi w sposób coraz bardziej skuteczny wdrożyć komputer do różnych inteligentnych zadań.

Udaje się to ze zmiennym szczęściem. Po początkowym entuzjazmie lat 50. nastąpił okres rozczarowania, gdy okazało się, że wydajność sprzętu jest niewystarczająca do wykonania zadań, które okazały się być dużo bardziej złożone, niż się spodziewano. Powrót nadziei nastąpił w latach 70. i 80., gdy możliwości sprzętu w znaczący sposób wzrosły i gdy opracowano nowe formalizmy i algorytmy pozwalające zmniejszyć złożoność wnioskowania. Zaczęły wtedy powstawać *systemy ekspertowe*, które znalazły pewne zastosowanie praktyczne. Spowodowały one rozwój m.in. komputerowych *baz wiedzy*, których głównym elementem były *ontologie*¹. Nie wnikając na razie zbyt głęboko w znaczenie tego terminu, ontologie są sporządzonymi z użyciem jakiegoś formalizmu logicznego opisami danej dziedziny. Jednak i tym razem fiasko wielu wiodących projektów badawczych doprowadziło do ograniczenia środków na finansowanie prac nad inteligentnymi systemami. Nastąpił wtedy okres nazywany czasami „zimą sztucznej inteligencji” (ang. *AI winter*).

Kolejny impuls kierujący zainteresowanie w stronę badań z zakresu *KRR* nastąpił po 2001 roku, kiedy to współtwórca Internetu, Tim Berners Lee, zaproponował nową inicjatywę o nazwie *Semantic Web*, która wskazywała drogę szerokiego zastosowania sztucznej inteligencji do przetwarzania informacji ukrytej w globalnej sieci WWW. Inicjatywa została przyjęta z entuzjazmem, gdyż w miarę wzrostu zasobów coraz trudniejsze było przeszukiwanie miliardów stron internetowych po to, by znaleźć te, które zawierają interesujące w danym momencie dane. Wydawało się, że inicjatywa jest celna, że tzw. znacznikowanie semantyczne i wykorzystywanie algorytmów wnioskujących, które potrafią odczytać i inteligentnie zinterpretować te znaczniki, pozwoli osiągnąć zamierzony cel. Jednakże, pomimo odnotowania znaczących osiągnięć, dziś wypada stwierdzić, że sukces jest jedynie połowiczny. Semantyczną warstwę opisową miały tworzyć ontologie. Powstaje ich wiele. Jednak jak dotąd nie rozpowszechniły się one na tyle, aby stać się praktycznie wykorzystywanym elementem Internetu. Potwierdzeniem tej opinii jest fakt, że nie powstała jeszcze przeglądarka, która potrafiłaby z nich skorzystać i w możliwym do zaaprobowania czasie

¹ Pojęcie bazy wiedzy (zwłaszcza jeśli chodzi o ich komputerową odmianę) jest szersze, niż pojęcie ontologii. W przypadku baz wiedzy opartych na logicznym opisie danej dziedziny, ontologia jest głównym ich składnikiem. Oprócz ontologii tymi składnikami są reguły i algorytmy wnioskowania oraz warstwa komunikacyjna (na przykład język zapytań). Jednak w ramach tej pracy oba pojęcia będą używane zamiennie.

odnaleźć trafną z semantycznego punktu widzenia odpowiedź. Na taki stan rzeczy składa się wiele przyczyn, a najważniejsze tkwią w samych właściwościach ontologii. Właściwości te wywołują cały szereg problemów na różnych etapach ich cyklu życia. Najczęściej wymienia się następujące:

- 1) Problemy występujące w czasie projektowania baz wiedzy:
 - a) *Problem integracyjny* polegający na niemożności efektywnego ponownego wykorzystania istniejących ontologii.
 - b) *Problem konstrukcyjny* polegający na braku możliwości zachowania kontroli nad kształtem modelu, czyli treścią tworzonego opisu.
 - c) *Problem organizacyjny* polegający na braku możliwości racjonalnego podziału prac związanych z rozwojem i utrzymaniem ontologii między wiele zespołów.
- 2) Problemy występujące w czasie eksploataowania baz wiedzy:
 - a) *Problem wydajnościowy* polegający na niskiej efektywności programów wnioskujących i innych systemów korzystających z baz wiedzy.
 - b) *Problem ochrony danych* polegający na niemożności zarządzania uprawnieniami dostępu do różnych części wiedzy.

Oczywiste jest, że problemy te w znacznym stopniu utrudniają tworzenie i użytkowanie ontologii, i sprawiają, że oba te etapy są bardzo kosztowne i angażują osoby o bardzo wysokim poziomie przygotowania merytorycznego, raczej naukowców niż inżynierów. Ze względu na ich wagę oraz znaczenie dla toku rozumowania zawartego w niniejszej pracy, problemy te będą dalej nazywane *problemami referencyjnymi*.

Podjęto wiele środków, które mogłyby opisanym wyżej problemom zaradzić. Jednym z tych środków była propozycja normalizacji tworzonych ontologii. Inicjatorzy tej propozycji wychodzili z założenia, że jeżeli wszyscy projektanci ontologii zastosują się do pewnych norm, tworzone przez nich ontologie staną się tańsze i o wiele bardziej zdatne do ponownego użycia. Druga propozycja dotyczyła modularyzacji ontologii, nazywanej też *kontekstualizacją*, gdyż moduły mają opisywać dany fragment wiedzy z jakiegoś punktu widzenia, w jakimś kontekście. Prace nad metodami modularyzacji skupiają się wokół sposobów odwzorowywania terminów z jednej ontologii na terminy z drugiej oraz na sposobach wnioskowania z wiedzy rozproszonej w wielu modułach. Obie propozycje wniosły bardzo wiele do dziedziny reprezentacji wiedzy. Z pewnością ustanowiły pewien docelowy, teoretyczny model, ku któremu powinny zmierzać rozwiązania stosowane w ramach inicjatywy Semantic Web. Model ten polega na dążeniu do utworzenia sieci powiązanych ze sobą ontologii, tworzących hierarchę, na której szczycie znajdowałyby się ontologie wysokiego poziomu (ang. *top-level ontologies*), opisujące wiedzę powszechną o świecie na wysokim poziomie abstrakcji, niżej umieszczone byłyby ontologie średniego poziomu (ang. *mid-level ontologies*), w końcu na najniższych szczeblach znajdowałyby się ontologie dziedzinowe (ang. *domain ontologies*), które wносиłyby już do sieci wiedzę szczegółową.

Zastosowanie różnych poziomów ontologii i modularyzacja pomogłyby rozwiązać wszystkie wymienione problemy referencyjne. Problem integracyjny zostałby rozwiązany dzięki temu, że po pierwsze, ontologie wysokiego i średniego poziomu byłyby wielokrotnie używane z definicji, po drugie, powtórne wykorzystanie różnych modułów dziedzinowych byłoby także możliwe, gdyż byłyby one oparte na wspólnej bazie ontologii wyższych poziomów, a algorytmy wnioskujące potrafiłyby poprawnie zachowywać się w środowiskach zmodularyzowanych. Wykorzystanie ontologii wysokiego i średniego poziomu rozwiązałyby również problem konstrukcyjny, gdyż inżynierowie zostaliby zwolnieni z konieczności

definiowania wielu pojęć ogólnych, skupialiby się więc jedynie na obszarze pojęć związanych z jakąś konkretną dziedziną. Ponadto mogliby, dzięki modularyzacji, dzielić tworzony model na mniejsze fragmenty, co pozwoliłoby łatwiej ogarnąć całość. Proponowane rozwiązanie złagodziłoby problem organizacyjny, gdyż poszczególne fragmenty można byłoby przydzielić różnym zespołom uczestniczącym w projekcie. Z kolei problem wydajnościowy zostałby rozwiązany dzięki możliwości wnioskowania lokalnego z pojedynczych modułów, a problem ochrony danych dzięki udzielaniu do tych modułów praw dostępu.

W praktyce model ten nie został jeszcze wdrożony i raczej trudno się spodziewać, że stanie się to prędko. Pomimo tego, że obie propozycje wydają się być trafne z teoretycznego punktu widzenia, w praktyce nie okazały się skuteczne. Przyczyna jest prosta: pomimo normalizacji, stosowania ontologii wysokiego poziomu i istnienia zaawansowanych sposobów łączenia modułów, tworzone ontologie wciąż są zbyt od siebie odległe w sposobie widzenia podobnych, wydawałoby się, problemów. Stąd wciąż tak samo trudno je łączyć i ponownie wykorzystywać. Z tego powodu przez ostatnich 10 lat warstwa ontologicznego opisu Internetu nie rozwinęła się w stopniu umożliwiającym wyraźny postęp w metodach inteligentnej eksploracji sieci.

Ostatnie zwiększenie zainteresowania oraz funduszy kierowanych na badania ontologii i sposobów ich wykorzystania zostało wywołane głównie przez inicjatywę Semantic Web. Jednak nie można powiedzieć, żeby Internet był jedynym miejscem ich używania. Wciąż najczęstszym miejscem zastosowania ontologii są systemy ekspertowe. Istnieją również skuteczne zastosowania w takich dziedzinach, jak integracja systemów i baz danych, przetwarzanie języka naturalnego, wydobywanie wiedzy, wspomaganie projektowania systemów informatycznych, itp. Realizacja niektórych zadań w ramach tych dziedzin byłaby bez ontologii wręcz niemożliwa. Są to jednak zastosowania bardzo specjalistyczne, dotyczą wąskich dziedzin, pojedynczych rozwiązań i najczęściej nie wychodzą poza etap badawczy. Istnieją także próby wiązania ontologii z systemami powszechnego użytku, na przykład z programami, w których stosuje się podejście kontekstowe (ang. *context-oriented programming*) ([GWPZ2004], [TSB2006]). Chodzi o to, aby programy te inteligentnie dostosowywały swoje działanie do profilu klienta i pozwoliły lepiej zaspokoić jego oczekiwania. Istnieją również próby przenoszenia informacji wprost z relacyjnych baz danych do baz wiedzy (na przykład w [CGY2007], [Z2008], [BC2009]) po to, aby można było czerpać z nich wiedzę możliwą do osiągnięcia jedynie dzięki zastosowaniu systemów wnioskujących. Jak dotąd próby te nie wyszły poza sferę eksperymentalną. Świadczą one jednak o tym, że pomimo opisanych wyżej barier istnieje silna wiara w słuszność takich rozwiązań.

U podstaw motywacji autora niniejszej rozprawy leży przekonanie, iż uda się opracować takie rozwiązanie, które sprawi, że możliwe stanie się szerokie zastosowanie baz wiedzy. Na przykład wplecenie technik sztucznej inteligencji w procesy biznesowe radykalnie podniosłoby ich funkcjonalność. Wyposażone w sztuczną inteligencję systemy wspomagające zarządzanie przedsiębiorstwem typu ERP, MRP, CRM, czy w końcu zwykłe systemy księgowo potrafiłyby to, co potrafią współczesne systemy tego typu, ale prócz tego, dzięki bogatszemu opisowi i umiejętności wnioskowania, mogłyby inteligentnie weryfikować działania użytkownika pod kątem prawnym lub ekonomicznym. Potrafiłyby przeanalizować uwarunkowania rynkowe i proponować przedsiębiorcy właściwe działania marketingowe. Dostosowałyby procedury związane ze sprzedażą do cech indywidualnych klientów. Pomogłyby racjonalnie zoptymalizować zasoby. Z kolei systemy związane z edukacją lub rozrywką dostosowywałyby się do potrzeb i upodobań indywidualnego użytkownika. Zastosowanie sztucznej inteligencji mogłoby ułatwić komunikację między różnymi systemami.

Główną przeszkodą w takim właśnie wykorzystaniu ontologii są opisane powyżej problemy referencyjne. Nieskuteczność dotychczas zastosowanych środków zaradczych wynika prawdopodobnie stąd, że nie została odkryta pierwotna przyczyna tych problemów. Zwolennicy normalizacji ontologii upatrują jej w zbytnej swobodzie twórców ontologii, zwolennicy modularyzacji w tym, że nie stosuje się dzielenia wiedzy na fragmenty i nie opisuje zależności między nimi. Autor niniejszej pracy uważa, że zwolennicy obu metod widzą tylko część problemu i właściwie nie trafiają w przyczynę najistotniejszą. Wyraził ją w [M1975] M. Minsky: „Wydaje mi się, że składniki większości teorii z dziedziny sztucznej inteligencji i psychologii są, ogólnie mówiąc, zbyt szczegółowe, zbyt lokalne i pozbawione struktury, by odzwierciedlić — praktycznie lub fenomenologicznie — efektywność zdroworozsądkowego myślenia. Cegiełki wnioskowania, języka, pamięci i percepcji powinny być większe i bardziej ustrukturalizowane; ich faktyczna i proceduralna zawartość musi być ściślej połączona, tak by wytłumaczyć ewidentną siłę i szybkość aktywności umysłowej”². Wydawać by się mogło, że idea modularyzacji wychodzi tej myśli naprzeciw. Tak jednak nie jest. Myśl Minsky’ego w efekcie doprowadziła do powstania bardzo dziś rozpowszechnionej metody tworzenia systemów informatycznych, nazywanej metodą obiektową. Pozwala ona ujmować w jeden obiekt fragment wiedzy o dowolnej wielkości. Fragment ten posiada semantykę, która jest integralną częścią semantyki opisywanego problemu. Jak dotąd formalizmy, które są wykorzystywane do tworzenia opisów ontologicznych, nie posiadają środków do tworzenia takich fragmentów. Semantyka proponowanych obecnie metod modularyzacji w żadnym stopniu nie dotyczy warstwy problemu, lecz metawarstwy, czyli warstwy opisu środków, które służą do opisu problemu. To nie jest to samo.

Idea Minsky’ego jest obecna także w innej, rozpowszechnionej dziś w inżynierii oprogramowania, metodzie — metodzie relacyjnej. Analogie między tą metodą, a metodą obiektową są powszechnie znane, więc nie ma potrzeby ich tu przytaczać. Ogólnie warto tylko powiedzieć, że relacja jest w ramach tej metody odpowiednikiem klasy obiektów, rozumianej jako opis typu i zbiór wystąpień. Ale metoda relacyjna posiada jedną ważną cechę, która sprawia, że jest ona w sferze strukturalizowania danych jeszcze bardziej użyteczna niż metoda obiektowa. Tą cechą jest możliwość łatwego tworzenia nowych, dowolnie określonych relacji w trakcie eksploatacji systemu. Sprawia to trafnie zdefiniowana algebra relacji. Metoda obiektowa takiej możliwości nie posiada. Klasy zdefiniowane na etapie projektowania systemu nie mogą już potem być zmieniane.

Autor niniejszej pracy jest przekonany, że brak wyżej diskutowanych cech jest przyczyną niskiej popularności ontologicznych metod opisu problemów i że istnieje możliwość poprawienia tego stanu rzeczy. Największe nadzieje budzi fakt, że istnieje wiele analogii między ontologicznym i relacyjnym (na etapie tworzenia modelu związków encji) lub obiektowym (na etapie tworzenia modelu klas) sposobem opisywania danej dziedziny. Możliwe jest więc zaproponowanie takich „cegiełek wnioskowania, języka, pamięci i percepcji”, które w analogiczny sposób pozwoliłyby wbudować w ontologię struktury, które obejmowałyby dowolnie duże fragmenty wiedzy i które byłyby interpretowane na poziomie opisu problemu (a nie na metapoziomie). Poważnym utrudnieniem jest jednak fakt, że struktury te musiałyby również umożliwiać odpowiednie wnioskowanie, analogiczne do wnioskowania w dotychczas tworzonych bazach wiedzy.

² W oryginale: *It seems to me that the ingredients of most theories both in Artificial Intelligence and in Psychology have been on the whole too minute, local, and unstructured to account—either practically or phenomenologically—for the effectiveness of common-sense thought. The chunks of reasoning, language, memory, and perception ought to be larger and more structured; their factual and procedural contents must be more intimately connected in order to explain apparent power and speed of mental activities.*

Niniejsza rozprawa ma na celu przedstawienie wyników badań nad zaproponowaną przez autora tej pracy metodą modularyzacji baz wiedzy, nazwanej metodą *SIM* (*Structural Interpretation Model*). Metoda ta jest propozycją wprowadzenia do ontologii budulca, który miałby opisane wyżej cechy. Zaproponowany budulec pozwala podzielić tworzony model na fragmenty dowolnej wielkości i nałożyć na nie określone związki, które zarówno odzwierciedlałyby strukturę opisywanego problemu, jak i określałyby odpowiednio przepływ wniosków.

Główna teza rozprawy brzmi:

Organizacja ontologii w formie hierarchii kontekstów powoduje uproszczenie ontologii i usprawnienie pracy systemów wnioskujących, a także ułatwia inżynierom ontologii konstruowanie złożonych baz wiedzy i bardziej adekwatne modelowanie w nich istotnych problemów występujących w inżynierii wiedzy, takich jak reprezentowanie różnych punktów widzenia, rozwiązywanie antynomii, aktualizowanie baz wiedzy przy zachowaniu monotoniczności i in.

Potrzebnych jest tu kilka słów wyjaśnienia. W tezie chodzi oczywiście o organizację ontologii zgodną z zasadami przedstawianej metody *SIM*. Z kolei zwrot „powoduje uproszczenie ontologii” jest pewnym skrótom myślowym, w którym chodzi o dostarczenie metod uporządkowania opisu wiedzy, wynikiem którego jest większa przejrzystość i czytelność ontologii. Nie chodzi więc tu o uproszczenie poprzez usunięcie części wiedzy, lecz poprzez nadanie jej struktury powodującej, że odczytanie zawartej w niej informacji staje się prostsze. Pragnienie nadania wiedzy odpowiedniej struktury zbliża proponowaną metodę do metod normalizacji i modularyzacji. Istnieje między nimi pewna wspólnota celów. Powoduje to, że niektóre z problemów referencyjnych to problemy, którym chce przeciwdziałać również metoda *SIM*. Na przykład uproszczenie ontologii i ułatwienie inżynierom konstruowania złożonych baz wiedzy wiąże się z usunięciem problemu konstrukcyjnego i organizacyjnego. Usprawnienie pracy systemów wnioskujących wiąże się z usunięciem problemu wydajnościowego. Ale istnieją też rozbieżności. Na przykład brak w tezie nawiązania do problemu integracyjnego i ochrony danych. Co prawda lepsza organizacja baz wiedzy pozwoli zapewne na łatwiejsze ponowne użycie, a struktura modułarna bardzo dobrze nadaje się do przydzielania uprawnień, jednak rozwiązanie tych problemów nie jest tu priorytetem. Pewne wątpliwości może też budzić ostatnia część tezy, mówiąca o adekwatności modelowania różnych punktów widzenia, w tym antynomii, i problemu aktualizacji danych powodujących utratę monotoniczności³, czyli inaczej mówiąc, o adekwatności modelowania kontekstowego. Niewątpliwie celem wszystkich metod modularyzacji jest umożliwienie modelowania kontekstowego⁴. Jednak adekwatność to coś więcej niż jedynie umożliwienie. Adekwatność oznacza też właściwe ujęcie związków między poszczególnymi punktami widzenia, a to jest dużo trudniejsze do zrealizowania.

Opisane rozbieżności wynikają przede wszystkim z różnych motywacji. Cele metody *SIM* są szerzej zarysowane. Nie chodzi w niej jedynie o dostarczenie odpowiednich narzędzi pozwalających zrealizować ideę semantycznego opisu informacji zawartej w Internecie. Chodzi w niej o to, aby w przyszłości uczynić z metody ontologicznej alternatywny w stosunku do metody relacyjno-objektowej sposób realizacji informatycznych systemów

³ Monotoniczność jest cechą wnioskowania, zgodnie z którą wszystkie wyciągnięte dotąd wnioski muszą pozostać prawdziwe. Ontologie powinny wspierać tę cechę, jednak w rzeczywistości trudno jest to osiągnąć, gdyż stoi to w sprzeczności z pragnieniem utrzymania aktualności opisu dziedziny, której stan zmienia się przecież w czasie. Dokładniej znaczenie monotoniczności jest opisane w punkcie 2.1.2.

⁴ Modelowanie kontekstowe ma na celu ujmowanie różnych punktów widzenia, w tym punktów widzenia powiązanych z różnymi momentami czasowymi, a więc mającymi związek z aktualizacją. Szerzej o modelowaniu kontekstowym w podrozdziale 4.1.

szerokiego użytku. To narzuca wymaganie wprowadzenia podziału wiedzy na takie fragmenty, które będą umożliwiały semantyczne ujęcie kontekstowe zarówno na poziomie ogólnym (terminologicznym), jak i na poziomie szczegółowym (opisu faktów). Innymi słowy, będą umożliwiały autentyczne *projektowanie kontekstowe*.

Osią przeprowadzanego wywodu będzie więc próba dowiedzenia, że zaproponowana metoda pozwala zaradzić niektórym z problemów referencyjnych, ale nacisk zostanie położony na to, że pozwala ona lepiej rozwiązać dodatkowe problemy:

- 1) *Problem adekwatności kontekstów* polegający na braku odpowiedniego budulca pozwalającego dzielić wiedzę na fragmenty dowolnej wielkości i posiadającego interpretację ulokowaną w ramach modelowanej dziedziny.
- 2) *Problem projektowania kontekstowego* polegający na braku odpowiednich metod pozwalających wyrazić w świadomy i zgodny z właściwościami danej dziedziny dobór punktów widzenia w celu jak najlepszego oddania jej charakterystyki

Dzięki wypunktowaniu problemów, które powinna rozwiązać metoda SIM, można w sposób bardziej precyzyjny przedstawić treść głównej tezy rozprawy, formułując następujące tezy pomocnicze:

Teza pomocnicza 1:

Metoda SIM pozwala rozwiązać problem konstrukcyjny, organizacyjny i wydajnościowy w podobny lub lepszy sposób, jak inne metody modularyzacji.

Teza pomocnicza 2:

Metoda SIM jako jedyna pozwala rozwiązać problem adekwatności kontekstów i problem projektowania kontekstowego.

Praca jest tak skonstruowana, aby kolejne jej rozdziały dostarczały wiedzy pozwalającej uzasadnić powyższe tezy. Rozdział 2 tłumaczy, czym są ontologie; przedstawia różne formalizmy logiczne służące do tworzenia ontologii oraz w jaki sposób ontologia jest powiązana z językiem i konceptualizacją; opisuje także miejsce ontologii w ramach inicjatywy Semantic Web. Rozdziały 3 i 4 są poświęcone metodom, których celem jest usunięcie lub złagodzenie problemów referencyjnych. Rozdział 3 dotyczy metod normalizacji, a rozdział 4 metod modularyzacji. Rozdział 5 przedstawia metodę SIM od strony formalnej, opisuje jej semantykę oraz nakreśla główne problemy wnioskowania. Zaproponowane zostały dwa algorytmy wnioskowania, których celem jest rozstrzygnięcie standardowych problemów wnioskowania z baz wiedzy. Zarysowane zostały też sposoby rozwiązywania niestandardowych problemów, które rozszerzają wachlarz możliwości korzystania z wiedzy, a które metoda SIM pozwala zastosować dzięki swoim właściwościom. Rozdział 6 przedstawia możliwości, które metoda SIM udostępnia w zakresie projektowania baz wiedzy. To właśnie te możliwości są podstawą uzasadnienia tezy pomocniczej nr 2. Rozdział ten zawiera wiele przykładów modelowania konkretnych sytuacji, w których semantyka struktury podziału bazy odzwierciedla właściwości opisywanej dziedziny. Rozdział 7 opisuje praktyczne zastosowanie metody SIM. Przedstawiono tam system wnioskujący, który został zaimplementowany w taki sposób, aby uwzględniać zasady narzucane przez opisywaną metodę. Przedstawiono również ontologię, która została zaprojektowana jako niezmodularyzowana i przetworzona według zasad metody SIM oraz rezultaty, jakie zostały dzięki temu osiągnięte. Rozdział 8 zawiera opis dalszych prac rozwijających ideę hierarchicznej kontekstualizacji ontologii, zarówno tych, które są obecnie prowadzone, jak i tych, których rozpoczęcie jest dopiero planowane. W końcu podsumowanie pracy zawiera ocenę metody, a tym samym uzasadnienie wyżej sformułowanych tez.

2 Ontologie w zastosowaniach informatycznych

Począwszy od Arystotelesa, poprzez kolejne okresy rozwoju nauki ludzkiej, aż do czasów współczesnych poprzedzających upowszechnienie się informatyki, ontologie były domeną filozofów. Były opisem wiedzy ludzkiej o rzeczach istniejących w świecie i służyły uporządkowaniu dyskusji naukowych pozwalających uzgadniać poglądy na temat tajemnicy bytu. Jako takie musiały być wyrażane ścisłym, eliminującym nieporozumienia językiem. Ich rozwój musiał więc wiązać się z rozwojem logiki i semantyki. Jednak całkowite wyeliminowanie nieporozumień mogło być możliwe dopiero po stworzeniu algorytmów pozwalających wszystkim w identyczny sposób interpretować treść zawartą w teoriach wyrażonych językami logiki. Potrzeba opracowania takich algorytmów pociągnęła za sobą powstanie idei maszyny Turinga, rozwój teorii automatów i w końcu doprowadziła do stworzenia komputerów. Jest więc naturalną rzeczą, że naukowcy wciąż dążą do zrealizowania marzenia o „mózgach elektronowych”, które potrafiłyby rozumować podobnie jak ludzie. W realizacji tego zadania pomocne są ontologie, jednak słowo to w tym wypadku jest używane w nieco innym, węższym znaczeniu.

Treść tego rozdziału określa tło dla dalszych rozważań, a także wnosi wiele ważnych pojęć, które będą intensywnie wykorzystywane w dalszej części rozprawy. Podrozdział 2.1 koncentruje się na definicji ontologii oraz wyjaśnieniu, w jaki sposób są one tworzone. Omawia również formę ontologii wykorzystywanych w klasycznych systemach ekspertowych. W podrozdziale 2.2 przedstawiono zastosowanie ontologii w ramach inicjatywy Semantic Web. Omówiono logikę opisową, która jest głównym formalizmem logicznym używanym w ramach tej inicjatywy, języki używane do wyrażania ontologii oraz stosowane systemy wnioskujące. Podrozdział 2.3 zawiera podsumowanie rozdziału.

2.1 Ontologie klasyczne

Znaczenie słowa *ontologia* zmieniało się w czasie. Początkowo oznaczało ono naukę o bycie, w późniejszym okresie oddzieliła się wersja znaczeniowa określająca teorię aksjomatyczną służącą do objaśniania znaczenia różnych terminów danego języka. W takim właśnie znaczeniu słowo to jest używane w niniejszej pracy. Do wyrażenia ontologii najlepiej nadają się języki oparte na formalizmach logicznych. Formalizmy te posiadają precyzyjnie zdefiniowaną semantykę i dzięki temu umożliwiają poprawną interpretację treści zawartej w aksjomatach. Semantyka ta pozwala również, przynajmniej w przybliżeniu, ująć formalnie coś, co w zasadzie wymyka się wszelkiej formalizacji — proces konceptualizacji.

Te właśnie sprawy są tematem niniejszego podrozdziału. W punkcie 2.1.1 opisano historyczną ewolucję znaczenia terminu *ontologia*. Punkt 2.1.2 opisuje w zarysie logikę pierwszego rzędu oraz podstawowe pojęcia związane z interpretacją oraz systemami dowodzenia. W punkcie 2.1.3 przedstawiono rozwinięcia logiki pierwszego rzędu, których celem jest ominięcie głównych ograniczeń logiki jako formy wyrażania ontologii, czyli nierozstrzygalności oraz wysokiej złożoności algorytmów wnioskowania. Punkt 2.1.4 ma na celu, z zastosowaniem definicji interpretacji logicznej, formalnie zdefiniować ontologię jako efekt procesu konceptualizacji.

2.1.1 Filozoficzne pojęcie ontologii

Słowo *ontologia* pochodzi ze starożytnej greki i oznacza naukę o bycie. Przez dłuższy czas było używane zamiennie ze słowem *metafizyka* i upowszechniło się dopiero w XVIII w. Pierwszy użył go Arystoteles, dla którego nauka o bycie stanowiła „naukę pierwszą”, czyli

najbardziej podstawową. Starożytni filozofowie wprowadzili szereg pojęć, które do dziś rzutują na sposób, w jaki opisywany jest wszechświat. Początkowo nauka nie rozdzielała rozważań filozoficznych od nauk opartych na doświadczeniu, stąd ontologiami nazywano zarówno prace opisujące byty idealne, jak i systematyzujące zjawiska zaobserwowane w przyrodzie. Według [H2004] ontologie można było wtedy podzielić na intuicyjne (oparte na wrodzonym ludziom „wglądzie” w istotę bytów), indukcyjne (uogólniające wyniki nauk szczegółowych) i aksjomatyczne (stosujące do opisu świata formuły logiczne). Wraz z utrwaleniem się w filozofii pragmatyzmu ontologie zaczęły się kojarzyć głównie z teoriami aksjomatycznymi, których celem jest analiza podstawowych terminów, pojęć lub założeń związanych z wiedzą obowiązującą w danej dziedzinie. Ludzki umysł, postrzegając zjawiska zachodzące w świecie rzeczywistym, radzi sobie z ich złożonością, dokonując *kategoryzacji*, *klasyfikacji* i *konceptualizacji* tych zjawisk. Proces ten polega na grupowaniu zjawisk podobnych, określaniu zależności między nimi oraz nadawaniu nazw zarówno grupom jak i zależnościom. Jest to więc działanie, które ma ścisły związek z kompetencjami językowymi człowieka. Z pojęciem ontologii wiąże się także pojęcie *przyporządkowania ontologicznego* (ang. *ontological commitment*), odnoszące się do zdolności danego języka (najczęściej chodzi o język formalny) do opisu wiedzy na temat stanu rzeczy w ramach danej dziedziny. Pojęcie to wprowadził Willard Quine w 1948 roku w artykule „On what there is” w czasopiśmie *Review of Metaphysics* i przedrukowanym w [Q1961]. Zasada przyporządkowująca brzmi: „istnieć, znaczy być podstawieniem zmiennej” (w oryginale: „*to be is to be the value of a variable*”). Innymi słowy należy przyjąć, że każdy termin istniejący w słowniku danego języka ma jakieś odniesienie w ramach dziedziny, do której opisu ten język został użyty. Przyczyną wprowadzenia takiego przyporządkowania jest chęć uniknięcia paradoksów wynikających z wartościowania zdań dotyczących bytów nieistniejących.

Ontologie są tworzone przy użyciu języków logiki. Jest to uzasadnione tym, że logika była przez wieki rozwijana właśnie jako nauka badająca powiązania między konceptualizacją, która jest dla ludzkich umysłów naturalnym sposobem radzenia sobie ze złożonością postrzeganego świata, a językiem, który jest najważniejszym środkiem komunikowania się ludzi. To powiązanie konceptualizacji z językiem opisuje semantyka języka. Narzuca ona sposób określania wartości logicznej (prawdy lub fałszu) zdań, a poszczególne terminy pozalogiczne odnosi do obiektów, zbiorów obiektów i relacji między obiektami opisywanej dziedziny. Ale logika daje jeszcze inny sposób określania wartości logicznej zdań. Dzięki zdefiniowaniu relacji wynikania pozwala mówić o tej wartości bez odniesienia do semantyki. Często bowiem można określić wartość logiczną zdania badając jego związek z innymi zdaniami. Ten związek określają reguły wnioskowania. Reguły te sprawiają, że, opisując jakieś zamierzone znaczenie, nie trzeba podawać wszystkich zdań, które je opisują. Dzięki temu ontologie nie muszą być nieskończonymi zbiorami zdań. Można podać jedynie pewien skończony podzbiór tych zdań. Reszta, dzięki stosowaniu reguł wnioskowania, może w każdej chwili zostać wywnioskowana.

2.1.2 Logika pierwszego rzędu jako język opisu wiedzy

Wśród wielu istniejących logik za punkt wyjścia można uznać *logikę pierwszego rzędu*, nazywaną też klasycznym językiem (rachunkiem) predykatów pierwszego rzędu. Powstała ona na przełomie XIX i XX wieku jako wynik prac wielu matematyków i filozofów, wśród których najbardziej znani to Gottlob Frege, Charles Pearce i Bertrand Russel. Od tego czasu jej pozycja ugruntowała się i w dniu dzisiejszym jest uznawana za logikę najbardziej znaną i najczęściej używaną. Często jest traktowana jako logika podstawowa, do której odnoszone są inne rodzaje języków logicznych. Z tego powodu warto w skrócie naszkicować jej cechy.

Z punktu widzenia *syntaktyki*, w logice pierwszego rzędu istnieją dwa rodzaje symboli: *logiczne* i *pozallogiczne*. Symbole logiczne to *spójniki logiczne* (\neg , \vee , \wedge , \exists , \forall), predykat identyczności $=$, nazwy *zmiennych* (najczęściej stosowane to x , y , z), i znaki interpunkcyjne (nawiasy okrągłe i przecinki). Symbole pozallogiczne to nazwy *funkcji* (np. *najlepszyPrzyjaciel*, a w rozważaniach ogólnych to najczęściej a , b , c , f , g) i *predykatów* (np. *StarszyNiż*, a w rozważaniach ogólnych najczęściej P , Q , R). Zarówno funkcje, jak i predykaty posiadają cechę, która nazywa się stopniem. Stopień jest liczbą naturalną określającą liczbę „argumentów”, które funkcja lub predykat może przyjąć. Na przykład funkcja *najlepszyPrzyjaciel*(x) jest stopnia 1, a predykat *StarszyNiż*(x , y) stopnia 2. Szczególne znaczenie posiadają funkcje i predykaty stopnia zerowego. Funkcja stopnia zerowego jest po prostu *stałą*, np. *janKowalski*. W tym przypadku nawiasy są pomijane. W rozważaniach ogólnych funkcje stopnia 0 są zazwyczaj oznaczane symbolami a , b , c , a funkcje stopni wyższych symbolami f , g . Z kolei predykaty stopnia zerowego nazywane są symbolami zdaniowymi.

W logice pierwszego rzędu istnieją dwa rodzaje wyrażeń: *termy* (nazywane też *argumentami*, np. [B2005]) i *formuły*. Termem jest:

- każda zmienna,
- wyrażenie postaci $f(t_1, \dots, t_n)$, gdzie f jest symbolem funkcji n -tego stopnia, a t_1, \dots, t_n są termami.

Formuły służą do budowania zdań. Można je budować w następujący sposób:

- Jeśli t_1, \dots, t_n są termami, a P symbolem predykatu n -tego stopnia, to $P(t_1, \dots, t_n)$ jest formułą.
- Jeśli t_1, t_2 są termami, to $t_1 = t_2$ jest formułą.
- Jeśli φ, ψ są formułami, a x jest zmienną, to konstrukcje $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $\forall x \varphi$ oraz $\exists x \varphi$ są formułami.

Formuły dwóch pierwszych rodzajów nazywane są *formułami atomowymi* lub *atomami*. Jeśli chodzi o formuły trzeciego rodzaju, formuły złożone, to stosowane są dodatkowe spójniki logiczne, które pozwalają uprościć niektóre wyrażenia:

- $\varphi \rightarrow \psi$ upraszcza formułę $\neg\varphi \vee \psi$.
- $\varphi \leftrightarrow \psi$ upraszcza formułę $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.⁵

Spójniki logiczne \exists i \forall (*kwantyfikator egzystencjalny* i *uniwersalny*) wiążą zmienne w formułach. Jeśli jakaś zmienna znajduje się poza zasięgiem kwantyfikatora, nazywana jest zmienną *wolną*. Formuły zdaniowe, które nie zawierają wolnych zmiennych, nazywane są *zdaniami*.

Łatwo zauważyć, że jeśli z syntaktyki logiki pierwszego rzędu usunięte zostaną termy i kwantyfikator oraz jeśli zbiór predykatów zostanie ograniczony tylko do predykatów stopnia zerowego, to wynikiem stanie logika zdaniowa. Logika zdaniowa jest więc podzbiorem logiki pierwszego rzędu.

Logika pierwszego rzędu swoją zdolność do opisu zjawisk obserwowanych w świecie rzeczywistym zawdzięcza trafnie zdefiniowanej semantyce. W latach 30. XX w. Alfred Tarski zaproponował w [T1937] interpretację opartą na teorii mnogości. Pomysł takiego

⁵ Równie dobrze można założyć, że zestaw podstawowych spójników logicznych to \neg , \rightarrow , \wedge , \exists , \forall . Wtedy spójnik \vee może być traktowany jako upraszczający, gdyż $\varphi \vee \psi$ upraszcza formułę $\neg\varphi \rightarrow \psi$.

właśnie interpretowania zdań jest powiązany z zasadą, która wynika z definicji prawdy, również autorstwa Tarskiego. Zasada ta mówi, że prawdziwość zdania sformułowanego w danym języku nie może być stwierdzona bez wychodzenia poza ten język. Aby więc określić, czy dane zdanie jest prawdziwe, należy wyjść poza ten język i przenieść się do innego systemu. Teoria mnogości modeluje świat w sposób zgodny z intuicją, gdyż pozwala operować zbiorami elementów będących odpowiednikami obiektów istniejących w rzeczywistości. *Interpretacja* jest więc w logice pierwszego rzędu definiowana jako para $(\Delta, \bullet^{\mathcal{I}})$, gdzie Δ jest niepustym zbiorem obiektów nazywanym *dziedziną*, a $\bullet^{\mathcal{I}}$ jest *przyporządkowaniem interpretacyjnym*, które:

- każdemu predykatowi n -tego stopnia P przyporządkowuje podzbiór n -tej potęgi kartezyjskiej zbioru Δ , co jest zapisywane $P^{\mathcal{I}} \subseteq \Delta^n$,
- każdej funkcji n -tego stopnia f przyporządkowuje n -arną funkcję $f^{\mathcal{I}}: \Delta^n \rightarrow \Delta$.

Aby opisać sposób interpretowania wyrażeń, należy jeszcze wprowadzić funkcję v wartościującą zmienne, która przypisuje im w ramach danej interpretacji elementy dziedziny. Dla danej interpretacji \mathcal{I} i funkcji v funkcję *denotacyjną* $v_{\mathcal{I}}$ definiuje się w następujący sposób:

- jeśli x jest zmienną, to $v_{\mathcal{I}}(x) = v(x)$,
- jeśli t_1, \dots, t_n są termami, a f jest funkcją n -tego stopnia, to $v_{\mathcal{I}}(f(t_1, \dots, t_n)) = f^{\mathcal{I}}(d_1, \dots, d_n)$, gdzie $f^{\mathcal{I}}: \Delta^n \rightarrow \Delta$, a $d_i = v_{\mathcal{I}}(t_i)$.

Interpretacja *przypisuje* formule *wartość logiczną prawdy* lub *falszu*. W logice zdaniowej przypisanie wartości logicznej odbywa się wprost poprzez funkcję wartościującą. W logice pierwszego rzędu decyduje o tym przyporządkowanie interpretacyjne. Czasem dla stwierdzenia, że formuła φ jest w interpretacji \mathcal{I} przypisana wartość prawdy używa się zapisu $\mathcal{I}[\varphi] = 1$. Mówi się, że interpretacja \mathcal{I} *spełnia* formułę φ lub że φ jest w \mathcal{I} *spełnione*. Relacja spełniania jest oznaczana symbolem \models .

Dla danej interpretacji \mathcal{I} i zgodnej z nią funkcji wartościującej v formuła φ jest spełniona, co jest zapisywane $\mathcal{I}, v \models \varphi$, gdy spełnione są następujące warunki:

Niech t_1, \dots, t_n będą termami, P predykatem stopnia n -tego, φ i ψ formułami, a x zmienną.

- 1) $\mathcal{I}, v \models P(t_1, \dots, t_n)$ wtwg $(d_1, \dots, d_n) \in P^{\mathcal{I}}$ oraz $d_i = v_{\mathcal{I}}(t_i)$;
- 2) $\mathcal{I}, v \models t_1 = t_2$ wtwg $d \in \Delta$ oraz $d = v_{\mathcal{I}}(t_1) = v_{\mathcal{I}}(t_2)$;
- 3) $\mathcal{I}, v \models \neg\varphi$ wtwg nie jest prawdą, że $\mathcal{I}, v \models \varphi$;
- 4) $\mathcal{I}, v \models (\varphi \wedge \psi)$ wtwg $\mathcal{I}, v \models \varphi$ oraz $\mathcal{I}, v \models \psi$;
- 5) $\mathcal{I}, v \models (\varphi \vee \psi)$ wtwg $\mathcal{I}, v \models \varphi$ lub $\mathcal{I}, v \models \psi$;
- 6) $\mathcal{I}, v \models \exists x \varphi$ wtwg $\mathcal{I}, v' \models \varphi$ dla jakiejś funkcji v' , która różni się od funkcji v co najwyżej wartościowaniem zmiennej x ;
- 7) $\mathcal{I}, v \models \forall x \varphi$ wtwg $\mathcal{I}, v' \models \varphi$ dla każdej funkcji v' , która różni się od funkcji v co najwyżej wartościowaniem zmiennej x ;

Jeśli dana formuła jest zdaniem, to jej wartość logiczna nie jest zależna od funkcji wartościującej v i można zapisać $\mathcal{I} \models \varphi$. Jeśli dana formuła jest spełniona we wszystkich interpretacjach, to jest ona *prawdziwa*. Taka formuła nazywa się *tautologią*. Jeśli istnieje co najmniej jedna interpretacja spełniająca daną formułę, to mówi się, że jest ona *spełnialna*.

Jeśli nie istnieje taka interpretacja, to formuła jest *niespełnialna*. Interpretacja spełniająca daną formułę nazywa się jej *modelem*.

Dany zbiór formuł S jest spełniony przez interpretację \mathcal{I} , co jest zapisywane $\mathcal{I} \models S$, wtwg wszystkie formuły należące do tego zbioru są spełnione przez tę interpretację. W takim przypadku interpretacja nazywa się *modelem* zbioru formuł S . Jeśli istnieje przynajmniej jedna interpretacja spełniająca dany zbiór formuł, to nazywa się go *zbiorem spełnialnym*, w przeciwnym przypadku jest on *zbiorem niespełnialnym*.

Celem stosowania logiki jest nie tylko sformalizowanie języka i jego semantyki, lecz przede wszystkim umożliwienie przeprowadzania w sposób nie budzący wątpliwości dowodów twierdzeń. Z przeprowadzaniem dowodów związane jest pojęcie konsekwencji logicznej. Formuła φ jest *konsekwencją logiczną* zbioru formuł S , co jest zapisywane $S \models \varphi$, jeśli we wszystkich interpretacjach, będących modelem S , formuła φ jest spełniona. Mówi się też, że φ *wynika* z S . Czyli każda interpretacja będąca modelem zbioru S musi też być modelem zbioru $S \cup \varphi$. Zapisując formalnie:

$$S \models \varphi \text{ wtwg } \forall \mathcal{I} (\mathcal{I} \models S \rightarrow \mathcal{I} \models S \cup \varphi)$$

Inaczej mówiąc, zbiór formuł $S \cup \neg\varphi$ jest wtedy niespełnialny. Gdy dana formuła wynika z pustego zbioru, co zapisuje się $\models \varphi$, to jest prawdziwa. Formuła jest także prawdziwa, gdy wynika z dowolnego zbioru formuł: $\forall S (S \models \varphi)$.

Poszukiwanie konsekwencji logicznych dla danego zbioru formuł nazywa się *wnioskowaniem*. Formuły, których logiczne konsekwencje są poszukiwane, nazywają się *przesłankami* wnioskowania, a formuła będąca ich logiczną konsekwencją nazywa się *konkluzją*. Wnioskowanie powinno spełniać trzy zasady tradycyjnie uważane za podstawowe: *zasadę zwrotności*, *zasadę idempotentności* i *zasadę monotoniczności*. Niech zbiorami przesłanek będą zbiory S , S_1 i S_2 . Niech funkcja f_C będzie funkcją wynikania, która zbiorowi S przyporządkowuje zbiór $f_C(S)$ wszystkich jego logicznych konsekwencji, czyli *zbiór konkluzji*.

- 1) Zasada zwrotności mówi, że $S \subseteq f_C(S)$. Oznacza to, że przesłanki wynikają same z siebie.
- 2) Zasada idempotentności mówi, że $f_C(f_C(S)) = f_C(S)$. Oznacza to, że formuły, które wynikają ze zbioru konkluzji, wynikają też ze zbioru przesłanek.
- 3) Zasada monotoniczności mówi, że $f_C(S_1) \subseteq f_C(S_1 \cup S_2)$. Oznacza to, że żadne wnioski wywiedzione z danego zbioru przesłanek nie tracą ważności po dodaniu do tego zbioru nowych przesłanek.

W celu sformalizowania procesu wnioskowania stworzonych zostało wiele różnych *systemów wnioskowania*, nazywanych też *systemami dedukcji* lub *systemami dowodzenia*. System dowodzenia składa się ze zbioru *aksjomatów* oraz zbioru *reguł dowodzenia*. Jego celem jest stworzenie *dowodu*, będącego opisem przebiegu pewnego wnioskowania wiodącego od zbioru aksjomatów do danej konkluzji. Opis ten powinien zawierać opis wszystkich kroków pośrednich oraz wskazywać użyte reguły. Nie każdy system potrafi dowieść istnienia relacji wynikania. Dlatego istnienie dowodu prowadzącego od zbioru przesłanek S do konkluzji φ zostało nazwane relacją *wyprowadzalności*, co jest zapisywane $S \vdash \varphi$. Aby zachować tożsamość między relacją wyprowadzalności a relacją wynikania, system musi posiadać cechy *pełności* i *poprawności*. System jest *pełny* wtedy, gdy jest w stanie wyprowadzić ze zbioru przesłanek każdą konkluzję, która z niego wynika: $\forall \varphi (S \models \varphi \rightarrow S \vdash \varphi)$. System jest *poprawny* wtedy, gdy każdą formułą, którą uda się wyprowadzić ze

zbioru przesłanek, jest ich logiczną konsekwencją: $\forall \varphi (S \vdash \varphi \rightarrow S \models \varphi)$. O systemie, który jest i pełny, i poprawny, mówi się, że jest *niezawodny* ([Z1996]).

Systemy dowodzenia można podzielić na *semantyczne* i *syntaktyczne*. Te pierwsze opierają się na znaczeniu formuł, czyli na ich właściwościach semantycznych. Te drugie abstrahują od semantyki i bazują na zasadzie mówiącej, że wnioskowanie musi wykorzystywać jedynie syntaktyczną strukturę formuł. Podejście syntaktyczne jest bardziej rozpowszechnione. W jego ramach dowód jest ciągiem zbiorów formuł takich, że każda formuła należąca do jednego z tych zbiorów jest albo aksjomatem, albo może być wyprowadzona z poprzednich formuł ciągu przy użyciu jednej z reguł dowodzenia. Ostatnim zbiorem w ciągu jest $\{\varphi\}$, gdzie φ jest wyprowadzonym twierdzeniem.

Najbardziej znanymi spośród klasycznych systemów dowodzenia są systemy Gentzena, Hilberta i system dedukcji naturalnej, który jest w pewnym sensie złożeniem obu pierwszych. Na marginesie warto dodać, że w stworzeniu tego ostatniego znaczący był wkład Polaków Łukasiewicza i Jaśkowskiego. Systemy te są często wykorzystywane w teoretycznych opracowaniach logicznych. Należą one do grupy systemów syntaktycznych. Oto kilka przykładowych reguł dowodzenia systemu dedukcji naturalnej:

$$\frac{\vdash \varphi, \vdash \varphi \rightarrow \psi}{\vdash \psi} \quad \text{(MP)}$$

$$\frac{\vdash \varphi, \vdash \psi}{\vdash \varphi \rightarrow \psi} \quad \text{(II)}$$

$$\frac{\vdash \neg \varphi \rightarrow \textit{fałsz}}{\vdash \varphi} \quad \text{(RA)}$$

$$\frac{\vdash \forall x \varphi(x)}{\vdash \varphi(x)} \quad \text{(UI)}$$

$$\frac{\vdash \varphi \rightarrow \psi(x)}{\vdash \forall x \psi(x)} \quad (x \text{ nie jest zmienną wolną w formule } \varphi) \quad \text{(UG)}$$

Reguła oznaczona symbolem (MP) jest nazywana *regułą odrywania*, choć często nazywa się ją łacińską nazwą *modus ponens*. Mówi ona, że jeżeli wyprowadzalna jest implikacja oraz jej poprzednik, to wyprowadzalny jest również następnik tej implikacji. Można więc „oderwać” go od implikacji i dołączyć do dowodu. Reguła (II) nazywa się *regułą wprowadzenia implikacji*, gdyż pozwala wstawić do dowodu implikację formuł uprzednio wyprowadzonych. *Reguła sprowadzenia do absurdu* (RA z łac. *reductio ad absurdum*) jest stosowana w dowodach przez zaprzeczenie. Należy zaprzeczyć udowodnianej tezie, a następnie wykazać, że prowadzi to do sprzeczności. Dotychczas omówione reguły dotyczyły logiki zdaniowej. Dwie następne reguły, *konkretyzacji* i *uniwersalnej generalizacji*, są przykładami reguł stworzonych dla logiki pierwszego rzędu. Pozwalają one pozbyć się z dowodu lub wprowadzić do niego formuły kwantyfikowane.

Wymienione wyżej klasyczne systemy dowodzenia nie nadają się do automatyzacji procesu wnioskowania, gdyż w ich ramach poszukiwanie właściwego dowodu jest zbyt trudne. W praktyce, a szczególnie w ramach zastosowań sztucznej inteligencji, bardzo ważnymi warunkami są rozstrzygalność i akceptowalny stopień złożoności wnioskowania. Warunki te spełnia opracowana w latach 60. XX w. ([R1965]) metoda wnioskowania zwana *rezolucją*. Opiera się ona na jednej tylko regule zwanej *regułą rezolucji*:

$$\frac{\vdash \varphi \vee \psi, \vdash \neg \varphi \vee \chi}{\vdash \psi \vee \chi} \quad \text{(RE)}$$

Łatwo zauważyć, że reguła ta uwzględnia tylko spójnik alternatywy. Ta właściwość reguły, a także pewne specyficzne cechy algorytmu rezolucji wymuszają, aby formuły tworzące ontologię miały postać tzw. *koniunkcyjnej postaci normalnej*. Formuła zapisana w tej postaci nazywa się *klauzulą*, dlatego postać ta jest inaczej nazywana *postacią klauzulową*. Aby stać się klauzulą, formuła musi być przekształcona w koniunkcję alternatyw formuł atomowych, przy czym negacje są dozwolone jedynie przed formułami atomowymi, kwantyfikatory egzystencjalne muszą być zastąpione funkcją (*skolemizacja*), a wszystkie kwantyfikatory uniwersalne muszą znajdować się na początku, w tzw. *prefiksie* klauzuli.

2.1.3 Rozwinięcia logiki pierwszego rzędu

Również rezolucja w niektórych przypadkach nie gwarantuje stopu, a w niektórych osiąga złożoność wykładniczą. Dlatego na potrzeby zautomatyzowanych systemów wnioskowania opracowano uproszczoną wersję tego algorytmu, zwaną *SLD-rezolucją*, która w znacznym stopniu te wady eliminuje. Wymaga ona jednak zmniejszenia ekspresywności języka, dopuszczając jedynie możliwość tworzenia tzw. *klauzul Horna* (od nazwiska ich pomysłodawcy), które mogą mieć tylko jedną *pozytywną* formułę atomową (pozytywną formułę atomową nazywa się formułę bez negacji). Klauzule Horna częściej nazywane są *regułami Horna* lub po prostu *regułami*.

Algorytm SLD-rezolucji oraz reguły Horna zainspirowały powstanie całego kierunku prac nad sztuczną inteligencją, który można określić *podejściem regułowym* (ang. *rule-based approach*). *Regułowe bazy wiedzy* składają się z dwóch części: części terminologicznej zawierającej reguły opisujące zależności między pojęciami w danej dziedzinie wiedzy oraz ze zbioru *faktów* zawierającego *ustalone formuły atomowe* (ang. *ground atoms*), czyli formuły, w których nie występują zmienne. Przykładem klauzuli z części terminologicznej może być formuła: $\forall x (PosiadaDowódOsobisty(x) \rightarrow Pełnoletni(x))$. Z kolei przykładem faktu jest formuła $PosiadaDowódOsobisty(janKowalski)$, która oznajmia, że obiekt oznaczony symbolem *janKowalski* spełnia predykat *PosiadaDowódOsobisty*. Tak utworzone bazy wiedzy pozwalają na tworzenie *regułowych systemów ekspertowych*, których celem jest udzielanie odpowiedzi na zapytania. Zapytanie jest także formułą ustaloną, a odpowiedź polega na stwierdzeniu, czy formuła ta jest prawdziwa.

W ciągu kilkudziesięciu lat rozwoju na bazie podejścia regułowego powstała dojrzała dziedzina, której celem jest tworzenie praktycznie wykorzystywanych regułowych baz wiedzy i systemów ekspertowych. Można wydzielić dwie gałęzie rozwoju tej dziedziny. Jedną z nich jest nazywana *programowaniem logicznym* (ang. *logic programming*). Jest ona oparta na językach takich jak Planner, Datalog, Prolog, R++ lub rodzina języków CLP. Najbardziej znany jest Prolog. W ramach programowania logicznego bazy wiedzy (reguły i fakty) są traktowane jak programy, które są *wykonywane* w interpreterze. Ten rodzaj programowania uzyskał nazwę *deklaratywnego*. Języki programowania deklaratywnego mają ekspresywność umożliwiającą twórcy programu sterowanie samym procesem jego wykonania, czyli pozwalającą wpływać na przebieg dedukcji. Drugą gałąź to tak zwane *regułowe systemy produkcyjne* (ang. *rule-based production systems*), zwane też *dedukcyjnymi bazami danych* (ang. *deductive databases*). W jej ramach nazwa „baza wiedzy” jest bardziej uzasadniona, gdyż zbiór reguł i faktów jest traktowany jak dane. Dane te stanowią bazę dla algorytmu, który z przesłanek produkuje konkluzje (stąd nazwa). Konkluzje te są najczęściej również zapamiętywane, czyli działanie systemu faktycznie przypomina funkcjonowanie baz danych. Oczywiście, jego celem jest także odpowiedź na pytanie o prawdziwość danej formuły, ale nie tworzy on dowodu, lecz próbuje tę formułę wyprodukować (lub odnaleźć ją wśród już wyprodukowanych). Przykładowymi regułowymi systemami produkcyjnymi są

Mycin, XCon, Soar oraz systemy wykorzystujące algorytm Rete, takie jak OPS5, Clips i SIPS2.

Często działanie systemów tworzonych w ramach dwóch opisanych gałęzi podejścia regułowego kojarzy się z pojęciami *wnioskowania w przód* (ang. *forward chaining*) i *wnioskowania wstecz* (ang. *backward chaining*). Istnieje tutaj pewne zamieszanie pojęciowe. Niektórzy autorzy (na przykład [M1998]) utożsamiają wnioskowanie wstecz z programowaniem logicznym i definiują je jako proces, który jest wykonywany po zadaniu zapytania (czyli gdy znany jest *cel* inferencji). Dlatego mówi się też, że jest to *wnioskowanie ukierunkowane na cel* (ang. *goal-directed reasoning*). Z kolei wnioskowanie w przód jest cechą systemów produkcyjnych i charakteryzuje się tym, że system jest gotowy produkować konkluzje zanim jeszcze pozna pytanie. W tym wypadku i przesłanki, i wyprodukowane konkluzje są pamiętane jako dane, stąd nazwa *wnioskowanie ukierunkowane na dane* (ang. *data-directed reasoning*). Inni autorzy ([BL2004]) podkreślają, że choć w praktyce rzeczywiście programowanie logiczne jest najczęściej związane z wnioskowaniem wstecz, a systemy produkcyjne z wnioskowaniem w przód, to jednak są to pojęcia niezależne. W ich ujęciu zarówno wnioskowanie w przód, jak i wnioskowanie wstecz są tylko i wyłącznie cechami mechanizmu tworzącego w ramach systemu dedukcyjnego dowód. Dowód, jako ciąg formuł łączący przesłanki z konkluzją, może być tworzony albo rozpoczynając od przesłanek i budując ciąg w stronę konkluzji, albo w kierunku odwrotnym — od konkluzji w stronę przesłanek. Zarówno programy logiczne, jak i systemy produkcyjne zdolne są wykorzystać i pierwszy, i drugi sposób budowania dowodu. Zawsze jednak musi być znana konkluzja, która ma być udowodniona. Jeśli więc system produkcyjny produkuje wszystkie możliwe wnioski, nie czekając na zapytanie (jak na przykład algorytm Rete), to nie jest możliwe zastosowanie wnioskowania wstecz⁶.

Wieloletnie doświadczenie doprowadziło do udoskonalenia algorytmów wnioskowania w taki sposób, aby z jednej strony uczynić efekty ich pracy bardziej przydatnymi w praktyce, z drugiej strony, aby przy minimalnym zmniejszeniu funkcjonalności uzyskać znaczącą redukcję złożoności obliczeniowej. Zmiany te spowodowały, że niektóre systemy regułowe nie mieszczą się już w logice pierwszego rzędu.

Jedną z najbardziej znaczących zmian jest wprowadzenie *negacji przez domniemanie* (ang. *negation as failure* — *NAF*). Zmiana ta powoduje, że w sytuacji, gdy nie jest możliwe dowiedzenie prawdziwości formuły φ , przyjmuje się, że prawdziwe jest zaprzeczenie tej formuły $\neg\varphi$. Byłoby to niedopuszczalne w klasycznym sposobie wnioskowania, gdzie wartość logiczna formuły φ pozostałaby nieznaną. Jednak w niektórych sytuacjach tak rozumiana negacja jest bardzo przydatna. Dotyczy to głównie sytuacji, gdzie kolekcja faktów zawiera pełną wiedzę o dziedzinie. Na przykład w przypadku, gdy baza wiedzy zawiera rozkład lotów, brak formuły *IstniejePołączenie(gdańsk, katowice)* (lub możliwości jej wywnioskowania) pozwala wysnuć z pewnością przydatny wniosek, że jest to zdanie nieprawdziwe, czyli że takie połączenie nie istnieje. Inną sytuacją, kiedy zasada NAF wykazuje swoją przydatność, jest sytuacja, w której istnieje reguła pozwalająca wyprodukować każdy element relacji spełniającej zaprzeczenie predykatu tworzącego formułę. Jeśli na przykład zdefiniowana jest reguła pozwalająca określić, czy dana liczba jest parzysta, nie trzeba już tworzyć nowej dla liczb nieparzystych, gdyż brak możliwości udowodnienia parzystości danej liczby gwarantuje jej nieparzystość.

W sytuacjach, gdy nie ma możliwości stworzenia takiej reguły, przyjęcie zasady NAF powoduje, że zbiór reguł i faktów stanowiących przesłanki jest traktowany jako pełna wiedza

⁶ W tym przypadku trudno jest też zakwalifikować tę produkcję jako wnioskowanie w przód, gdyż konkluzje, które należy udowodnić, są nieznanne.

o dziedzinie. Jest to więc odejście od przyjętego w logice klasycznej założenia o świecie otwartym (ang. *open world assumption* — OWA), które mówi, że pełna wiedza o świecie nie jest znana, w stronę założenia o świecie zamkniętym (ang. *closed world assumption* — CWA) przyjmującego, że zbiór przesłanek jest skończony i algorytm wnioskujący ma do niego dostęp. Negacja przez domniemanie jest też odejściem od jednej z klasycznych zasad wnioskowania, mianowicie od zasady monotoniczności. Jeśli wskutek braku możliwości dowiedzenia prawdziwości formuły φ zostanie wyciągnięty wniosek $\neg\varphi$, to po dodaniu nowych formuł, z których da się jednak dowieść prawdziwości φ , trzeba będzie formułę $\neg\varphi$ usunąć ze zbioru konkluzji. Tak więc zależność $f_C(S_1) \subseteq f_C(S_1 \cup S_2)$ nie zostanie zachowana.

Inną zmianą, która oddala regułowe bazy wiedzy od logiki pierwszego rzędu, jest odejście od *monotoniczności zbioru przesłanek*, czyli umożliwienie wycofywania wprowadzonych do nich wcześniej reguł i faktów. Jest to możliwość, która wychodzi naprzeciw potrzebie uwzględnienia *dynamiki* opisywanej dziedziny. W każdej dziedzinie zachodzą zmiany, a więc z łatwością może dojść do sytuacji, kiedy wprowadzone wcześniej do bazy wiedzy fakty (a nawet reguły) mogą stać się fałszywe. Z punktu widzenia logiki pierwszego rzędu nie jest to zmiana stanu, lecz przejście do zupełnie nowego zbioru przesłanek.

Udostępnienie programistom korzystającym z języków programowania logicznego kontroli nad przebiegiem wnioskowania prowadzi do utraty pełności i poprawności ([L1987]). Na przykład w języku Prolog istnieje *operator odcięcia* (ang. *cut operator*), który pozwala twórcy programu nakazać interpreterowi pominięcie niektórych ścieżek wnioskowania.

Systemy regułowe osiągnęły wysoką sprawność. Na przykład algorytm Rete [F1982] osiąga złożoność subliniową w stosunku do liczby wprowadzonych faktów. To sprawia, że stały się one potężnym narzędziem pozwalającym inżynierom wiedzy rozwiązywać wiele problemów z zakresu sztucznej inteligencji. Jednak mają też szereg wad. Odejście od klasycznych zasad wnioskowania, mimo że poprawiło parametry działania, sprawiło, że zawężił się obszar ich zastosowań. Na przykład założenie o świecie zamkniętym sprawiło, że wykluczony został cały obszar tych zastosowań, gdzie z góry wiadomo, że wiedza jest niepełna. Inną wadą regułowych baz wiedzy jest ich *płatkość*: składają się one z małych elementów, reguł i faktów, które mogą być rozumiane niezależne od siebie. Informacja o danym obiekcie może występować w bardzo wielu pozornie niezależnych od siebie formułach i nie istnieje żaden sposób, aby te formuły semantycznie pogrupować. Stoi to w sprzeczności z wspomnianą już naturalną umiejętnością ludzkiego umysłu — z umiejętnością kategoryzacji. Płatkość bazy powoduje też to, że w miarę wzrostu liczby reguł narasta trudność z ogarnięciem zawartej w niej wiedzy.

W ostatnich latach coraz większą popularnością cieszy się inny kierunek rozwoju baz wiedzy, kierunek oparty na tzw. *logice opisowej* (ang. *Description Logics* — DL). Intencją pomysłodawców tej logiki było stworzenie formalizmu, który byłby przydatny do opisu systemów ramkowych oraz sieci semantycznych. Ideę systemu ramkowego zaproponował w M. Minsky ([M1975]) w latach 70. ubiegłego wieku i stała się ona podstawą dla systemów obiektowych. Z kolei prace nad sieciami semantycznymi zostały zainicjowane dekadę wcześniej w rozprawie doktorskiej przez Quiliana (omówienie w [Q1966]). Oba nurty badań ukierunkowane były na stworzenie metod opisu dziedziny uwzględniających możliwość grupowania obiektów w ramach kategorii (klas), przypisywania im cech grupowych oraz dziedziczenia tych cech. Logika opisowa ujmuje te właściwości dziedziny, pozwalając opisywać ją jako graf, w którego węzłach znajdują się obiekty, a krawędziami są powiązania między tymi obiektami. Do opisu tak rozumianej dziedziny wystarczą jedynie predykaty unarne i binarne. Jest to silne ograniczenie ekspresywności w stosunku do logiki pierwszego

rzędu, jednak umożliwia zachowanie rozstrzygalności i ogranicza złożoność. Zastosowany tu system wnioskujący nie opiera się na rezolucji, lecz na algorytmie *tableau*, który przekształca formuły tworzące bazę wiedzy w graf. Jest to cecha, która sprawia, że zastosowany system powinien zostać zaliczony do systemów semantycznych. Jednakże należy podkreślić, że logika opisowa nie wykracza poza zakres logiki pierwszego rzędu (jest jej podzbiorem).

Najsilniej podkreślaną cechą logiki opisowej jest jej zdolność do tworzenia specyficznych *fraz rzeczownikowych* (ang. *noun phrases*), dzięki którym prosto i bez utraty precyzji da się opisać przynależność obiektu. Uzyskuje się ją dzięki możliwości tworzenia *ad hoc* nowych nazw predykatów, odpowiednio składając nazwy istniejące. Na przykład, aby powiedzieć, że ktoś jest pracowitym studentem, można napisać formułę atomową $Pracowity \sqcap Student(janKowalski)$, co w logice pierwszego rzędu ma swój odpowiednik w postaci formuły złożonej $Pracowity(janKowalski) \wedge Student(janKowalski)$. Taki zapis umożliwia tzw. *wnioskowanie terminologiczne*, czyli orzekanie, w jakiej relacji są względem siebie ekstensje⁷ dwóch danych predykatów: czy są w relacji subsumcji⁸, czy posiadają wspólną część, itp. Jest to wielka siła tego systemu.

Logika opisowa jest w niniejszej rozprawie traktowana jako formalizm wyjściowy do opisu proponowanej metody modularyzacji baz wiedzy. Dlatego też zostanie jej poświęcony odrębny rozdział (2.2.2). W tym miejscu warto jedynie wskazać na główne różnice między systemami regułowymi a systemami opartymi na logice opisowej. W skrócie najważniejsze różnice są następujące.

Logika opisowa:

- Zachowanie zasad klasycznego wnioskowania (monotoniczność, OWA).
- Możliwość wnioskowania terminologicznego.
- Czytelny i zrozumiały model wykorzystujący ludzką zdolność do kategoryzacji.
- Łatwiejsze powtórne wykorzystanie.

Systemy regułowe:

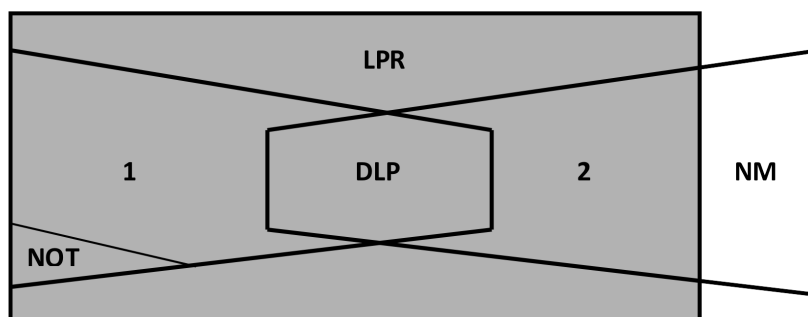
- Lepsza kombinacja ekspresywności i złożoności.
- Sprawne i sprawdzone metody wnioskowania.
- Większa dojrzałość dzięki szerokiemu zastosowaniu praktycznemu.

Istnieją próby stworzenia nowych systemów, które łączyłyby w sobie zalety obu podejść. Są one utrudnione, gdyż proste połączenie wprowadza nierozstrzygalność. Prawdopodobnie najbardziej udaną próbą jest metoda nazwana *programowaniem DL* (ang. *Description Logic Programming — DLP*) ([GHVD2003]). Na rysunku 2.1 przedstawiono wzajemną zależność różnych systemów. Szary prostokąt symbolizuje logikę pierwszego rzędu. Całkowicie mieści się w nim logika opisowa ($DL = 1 + NOT + DLP$). Skrót SLD

⁷ Najczęściej stosowaną nazwą używaną do określenia zbioru obiektów (lub krotek), które opisuje nazwa predykatu, jest słowo *zakres*. W niniejszej rozprawie będzie ono zastępowane przez rzadziej stosowaną *ekstensję* lub *odniesienie przedmiotowe*. Jest to spowodowane tym, że w ramach logiki opisowej słowo *zakres* jest używane do określenia przeciwdziedziny roli.

⁸ Słowo *subsumcja* (*subsumpcja*) jest często używane zamiennie ze słowem *dziedziczenie*. W niniejszej pracy pierwszy termin jest używany w sytuacjach, w których ekstensja jednego predykatu mieści się w ekstensji drugiego (czyli w sytuacjach, w których wzajemna relacja między predykatami jest określana na podstawie ich semantyki). Drugi termin jest używany w sytuacjach, w których chodzi o przenoszenie się cech (właściwości, zachowania), na przykład może odnosić się do zależności między klasami w metodzie obiektowej.

symbolizuje systemy oparte na regułach Horna, czyli na SLD-rezolucji ($SLD = DLP + 2$). Techniki oznaczone skrótowo jako niemonotoniczne (NM) wychodzą poza zakres logiki pierwszego rzędu. Programowanie DL (DLP) jest wspólną częścią logiki opisowej i reguł Horna.



LPR – logika pierwszego rzędu
 NOT – klasyczna negacja
 DLP – Description Logic Programming
 NM – niemonotoniczność

$$DL = 1 + NOT + DLP \quad SLD = 2 + DLP \quad LP = SLD + NM$$

Rys. 2.1 Wzajemna zależność między różnymi systemami opisu wiedzy, bazującymi na logice.

2.1.4 Formalizacja pojęcia ontologii

Powyższe przedstawienie różnych systemów logicznych umożliwia bardziej formalne odniesienie się do pojęć z początku rozdziału, takich jak ontologia, konceptualizacja, przyporządkowanie ontologiczne języka.

Najbardziej znaną definicją ontologii jest definicja T. Grubera [G1995], w której określa się ontologię jako *specyfikację konceptualizacji*. Słowo *specyfikacja* kojarzy się z precyzyjnie wyrażoną listą, w tym wypadku listą elementów, części składowych konceptualizacji. Aby jednak lepiej zrozumieć dzięki tej definicji sens słowa ontologia, należy najpierw sformalizować pojęcie konceptualizacji. Sam Gruber, używając tego terminu powoływał się na [GN1987], gdzie konceptualizacja została zdefiniowana jako struktura (Δ, R) , gdzie Δ jest dziedziną, a R zbiorem określonych na niej relacji. Jednakże N. Guarino w [G1998] skrytykował to podejście, zarzucając mu ekstensjonalność, podczas gdy konceptualizacja jest zjawiskiem intensjonalnym. Mówiąc *ekstensjonalność*, Guarino miał na myśli matematyczne rozumienie relacji (elementu zbioru R) jako podzbioru Δ^n , gdzie n jest stopniem relacji. *Intensjonalność* z kolei jest rozumiana jako powiązanie myśli czy idei, do której odnosi się dany termin, ze zjawiskiem obserwowanym w świecie rzeczywistym. *Relacje intensjonalne*, które Guarino później nazywa *relacjami konceptualnymi* (ang. *conceptual relations*), w przeciwieństwie do zwykłych relacji matematycznych odbijają pewien konkretny stan rzeczy. Na przykład w świecie klocków leżących na stole słowo *nad* odnosi się do sposobu, w jaki ma się do siebie przestrzenne ułożenie dwóch elementów. Relacje konceptualne są zdefiniowane na *przestrzeni dziedzinowej* (ang. *domain space*), którą można opisać strukturą (Δ, W) , gdzie Δ jest dziedziną, a W zbiorem *światów możliwych*, czyli *maksymalnych stanów rzeczy* w tej dziedzinie. We wspomnianym świecie klocków na stole zbiorem światów możliwych będzie zbiór wszystkich możliwych ułożeń tych klocków. W tak zdefiniowanej przestrzeni dziedzinowej można zdefiniować relację konceptualną jako funkcję $r^n: W \rightarrow \mathcal{P}(\Delta^n)$ ze zbioru światów możliwych do zbioru wszystkich n -arnych relacji

(matematycznych) na Δ . Dla ogólnej relacji konceptualnej r , zbiór $\mathcal{E}_r = \{r(w) \mid w \in W\}$ będzie zawierał wszystkie *dopuszczalne ekstensje* (ang. *admittable extensions*) r . Można już zatem zdefiniować konceptualizację.

Definicja 2.1 (konceptualizacja)

Konceptualizacja C jest trójką (Δ, W, \mathcal{R}) , gdzie \mathcal{R} jest zbiorem relacji konceptualnych zdefiniowanych na przestrzeni dziedzinowej (Δ, W) . ■

Konceptualizacja jest w tym ujęciu zjawiskiem znacznie bogatszym niż ta, która wynika z definicji z [GN1987]. Struktura (Δ, R) odnosi się do jednego stanu rzeczy. Dlatego też można ją nazwać *strukturą świata* (ang. *world structure*). Dana konceptualizacja zawiera wiele takich struktur, po jednej dla każdego świata możliwego. Guarino nazywa je *zamierzonymi strukturami świata* (ang. *intended world structures*) dla tej konceptualizacji.

Definicja 2.2 (zamierzona struktura świata)

Dla każdego świata możliwego $w \in W$ i dla konceptualizacji $C = (\Delta, W, \mathcal{R})$ *zamierzona struktura świata* w jest strukturą $S_{wC} = (\Delta, R_{wC})$, gdzie $R_{wC} = \{r(w) \mid r \in \mathcal{R}\}$ jest zbiorem ekstensji elementów \mathcal{R} dla świata w . Symbol S_C oznacza zbiór $\{S_{wC} \mid w \in W\}$ wszystkich zamierzonych struktur świata dla C . ■

Aby precyzyjnie wyrazić, czym jest ontologia, Guarino używa także pojęcia przyporządkowania ontologicznego. Wychodzi od zwykłego pojęcia interpretacji dla danego języka logicznego L ze słownikiem V , Guarino nieznacznie modyfikuje jego definicję. Modelem dla języka L jest para (S, I) , gdzie S jest strukturą świata (Δ, R) , a I jest przyporządkowaniem interpretacyjnym $V \rightarrow \Delta \cup R$. Dzięki tej modyfikacji wyróżnione zostały w słowniku nazwy indywidualne, którym funkcja I przyporządkowuje elementy dziedziny Δ , oraz predykaty, którym przyporządkowuje elementy R . Model wyznacza ekstensjonalną interpretację języka. W przypadku konceptualizacji, która jest zjawiskiem intensjonalnym, potrzebne jest pojęcie interpretacji intensjonalnej, czyli właśnie przyporządkowania ontologicznego. Jest ono zdefiniowane w ten sposób.

Definicja 2.3 (przyporządkowanie ontologiczne)

Przyporządkowaniem ontologicznym \mathcal{K} dla języka L o słowniku V jest para (C, \mathcal{J}) , gdzie C jest konceptualizacją (Δ, W, \mathcal{R}) , a \mathcal{J} jest funkcją $\mathcal{J}: V \rightarrow \Delta \cup \mathcal{R}$, przyporządkowującą nazwom indywidualnym ze słownika V elementy dziedziny Δ , a nazwom predykatów z tego słownika relacje konceptualne ze zbioru \mathcal{R} . ■

Ostatnim elementem potrzebnym do zdefiniowania pojęcia ontologii jest pojęcie zbioru *zamierzonych modeli* (ang. *intended models*).

Definicja 2.4 (zbiór zamierzonych modeli)

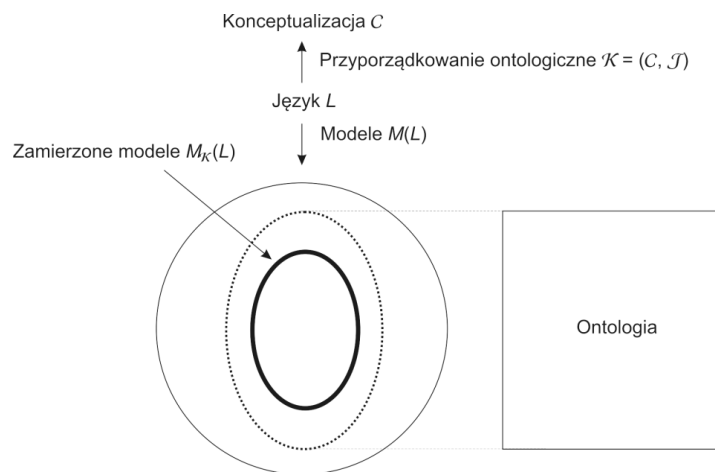
Zbiorem *zamierzonych modeli* $M_{\mathcal{K}}(L)$ języka L o słowniku V ze względu na przyporządkowanie ontologiczne $\mathcal{K} = (C, \mathcal{J})$ jest zbiór zawierający takie modele (S, I) , że:

- 1) $S \in S_C$,
- 2) dla każdej nazwy indywidualnej $a \in V$ $\mathcal{J}(a) = I(a)$,
- 3) istnieje taki świat możliwy $w \in W$, dla którego każdy symbol predykatu $P \in V$ funkcja I odwzorowuje w dopuszczalną ekstensję $\mathcal{J}(P)$, inaczej mówiąc, istnieje konceptualna relacja $r \in \mathcal{R}$, taka że $\mathcal{J}(P) = r$ oraz $r(w) = I(P)$. ■

Wyposażony w powyżej zdefiniowane pojęcia, Guarino postuluje, aby sprecyzować pojęcie ontologii rozumianej jako specyfikacja konceptualizacji:

Ontologia jest teorią logiczną opisującą *zamierzone znaczenie* (ang. *intended meaning*) formalnego słownika, tj. przyporządkowanie ontologiczne tego słownika do danej konceptualizacji świata. Zbiór zamierzonych modeli języka logicznego, używającego tego słownika, jest ograniczany przez jego przyporządkowanie ontologiczne. Ontologia w sposób pośredni odzwierciedla to przyporządkowanie (i kryjącą się za nim konceptualizację) poprzez przybliżenie tych zamierzonych modeli.

Jeśli więc mamy język L i przyporządkowanie ontologiczne \mathcal{K} , to ontologia O jest zbiorem aksjomatów wyrażonych w tym języku, dobranych w ten sposób, aby zbiór jej modeli w maksymalnym stopniu pokrywał zbiór zamierzonych modeli tego języka ze względu na to przyporządkowanie. Zależności te ilustruje rysunek 2.2.



Rys. 2.2 Rola ontologii w wyrażaniu zamierzonych modeli języka dla danej konceptualizacji (za [G1998])

Zamierzone modele języka L ze względu na przyporządkowanie ontologiczne \mathcal{K} reprezentowane są przez punkty wewnątrz pogrubionej elipsy. Są one podzbiorem zbioru modeli ontologii reprezentowanego przez elipsę narysowaną linią punktową. Obie elipsy znajdują się wewnątrz okręgu reprezentującego wszystkie modele wyznaczone przez język L .

Ontologia jest zatem zbiorem zdań (aksjomatów) wyrażonych w danym języku, których semantyka opisuje (nieodkładnie) jakąś konceptualizację. Jeśli język jest bardziej ekspresywny, to opis konceptualizacji jest dokładniejszy, jednakże można założyć, że w pełni dokładny opis nie jest możliwy.

2.2 Ontologie w Internecie

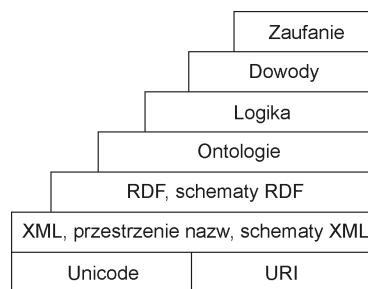
Rozwój Internetu, a konkretnie stopniowa utrata kontroli nad jego zawartością, sprawił, że ponownie zainteresowano się ontologiami. Tym razem chodziło o opis świata Internetu. Opis ten pozwoliłby lepiej uporządkować zawartość stron WWW, a nadanie tej zawartości semantyki pozwoliłoby w inteligentny sposób z niej korzystać. Ten wzrost zainteresowania zaowocował dużym postępem w propagowaniu ontologicznego sposobu przechowywania i przetwarzania informacji. Opracowano ogólnoswiatowe standardy w zakresie języków i sposobu zapisu formuł oraz uzyskano znaczący postęp w wydajności algorytmów wnioskujących.

Celem tego podrozdziału jest skrótowe przedstawienie założeń inicjatywy Semantic Web oraz standardów wykorzystywanych w jej ramach. Szczególną uwagę poświęcono logice

opisowej, gdyż stanowi ona podstawowy formalizm używany w niniejszej rozprawie do zapisu ontologii. W punkcie 2.2.1 opisano w skrócie założenia opisanej inicjatywy. Punkt 2.2.2 przedstawia charakterystykę logiki opisowej. Charakterystyka ta koncentruje się przede wszystkim wokół definicji pojęć, które będą używane w dalszej części rozprawy. Punkt 2.2.3 przedstawia najważniejsze języki oraz systemy wnioskujące używane w ramach inicjatywy Semantic Web.

2.2.1 Założenia inicjatywy Semantic Web

Za twórcę pojęcia Semantic Web uważa się Tima Bernersa-Lee, autora podstawowych założeń sieci WWW i dyrektora konsorcjum W3C, którego celem jest propagowanie i ustalanie standardów związanych z dobrym wykorzystaniem Internetu. Idea Semantic Web [BHL2001] polega na uzupełnianiu dostępnych w globalnej sieci danych o pewne metainformacje, dzięki którym znaczenie tych danych staje się, mówiąc metaforycznie, zrozumiałe dla komputera. Technologie, które oferuje lub wykorzystuje Semantic Web, zorganizowane są w warstwy formujące tzw. tort semantycznego Internetu. Jedną z graficznych wersji tego tortu przedstawiono na rysunku 2.3.



Rys. 2.3 Warstwowa organizacja standardów i technologii wykorzystywanych przez Semantic Web (za <http://www.semanticweb.org>)

Jak widać na rysunku, Semantic Web bazuje na uznanych standardach używanych w sieci, takich jak *XML*. Dwie pierwsze (patrząc od dołu) warstwy obejmują właśnie takie technologie. Rekomendacją W3C związaną już bezpośrednio z Semantic Web jest *RDF* (ang. *Resource Description Framework*), czyli język opisu zasobów. Język *RDF* definiuje specyficzną konstrukcję zdaniową zwaną trójką *RDF* (ang. *RDF triple*). Za pomocą trójek możliwe jest ustanawianie zależności pomiędzy zasobami sieciowymi lub reprezentowanymi przez nie encjami. Kolejna warstwy struktury obejmuje ontologie. Warstwa ta jest bardzo ważna z punktu widzenia możliwości integracji i unifikacji informacji znajdującej się w Internecie, a także bardzo ściśle zintegrowana z sąsiednią warstwą *logiki*. Zasady logiczne zastosowane w ontologiach pozwalają na wnioskowanie, co oznacza, że przetwarzane dane wzbogacane są o semantykę i możliwe jest ich uzupełnianie o wnioski z nich wynikające. W bardziej zaawansowanych zastosowaniach ontologii, wymagających wysokiego stopnia integralności (czyli przede wszystkim tam, gdzie liczy się bezpieczeństwo), ważne może okazać się dostarczanie i weryfikowanie poszczególnych kroków dowodów logicznych. Takie mechanizmy, połączone z użyciem dodatkowych środków takich, jak np. ścieżki certyfikacji, stanowią mogą ważny element infrastruktury budującej *zaufanie* do systemów Semantic Web.

2.2.2 Logika opisowa

Termin logika opisowa [BMNP2003] zwykle stosuje się do rodziny formalizmów logicznych charakteryzujących się pewnymi wspólnymi cechami, natomiast w nieco szerszym

znaczeniu, oznacza dziedzinę nauki zajmującej się badaniem właściwości takich formalizmów. Najważniejsze wspólne cechy, które decydują o stosunkowej jednolitości tego obszaru badań, to:

- wykorzystywanie jako podstawy logiki pierwszego rzędu: formalizmy logiki opisowej są w ogromnej większości podzbiorami tej logiki, czyli, co za tym idzie, ograniczają ekspresywność klasycznego rachunku predykatów,
- nacisk na właściwości obliczeniowe: wykorzystywane formalizmy logiki opisowej są dobierane tak, aby wykazywały pożądane właściwości obliczeniowe, przede wszystkim rozstrzygalność,
- jednolity model dziedziny: dziedzina traktowana jest jako zbiór *osobników* (ang. *individuals*) należących do ekstensji tzw. *konceptów* (ang. *concepts*), połączonych wystąpieniami *ról* (ang. *roles*). Koncepty stanowią odpowiedniki predykatów unarnych, a role binarnych — predykaty wyższych stopni nie są zazwyczaj wykorzystywane,
- nacisk na tworzenie tzw. *deskrypcji*, czyli fraz rzeczownikowych: w formalizmach logiki opisowej często zamiast złożonych wyrażeń wykorzystuje się złożone predykaty formułowane za pomocą tzw. *konstruktorów*, np.:
 $Pracowity \sqcap Student(janKowalski)$
zamiast $Pracowity(janKowalski) \wedge Student(janKowalski)$.

Szczególnie ważny z punktu widzenia zastosowania w ramach inicjatywy Semantic Web jest nacisk na rozstrzygalność. Wybranie logiki opisowej jako podstawy dla reprezentowania wiedzy zawartej w Internecie gwarantuje, że przeprowadzanie wnioskowania w sposób poprawny i pełny będzie wykonalne (choć jego złożoność bywa w przypadku wielu formalizmów bardzo wysoka).

Logika opisowa zostanie przedstawiona w dalszej części niniejszego punktu na przykładzie formalizmu *SHOIQ*, którego ekspresywność zbliżona jest do ekspresywności języka OWL (semantyka bezpośrednia) wykorzystywanego w ramach inicjatywy Semantic Web do zapisu ontologii (patrz punkt 2.2.3). Formalizm ten wykorzystuje stosunkowo szeroką gamę konstruktorów konceptów. Konstruktory te zaprezentowane są w poniższej definicji.

Definicja 2.5 (konstruktory konceptów formalizmu *SHOIQ*)

Dla dowolnego konceptu atomowego (nazwy generalnej) A , dowolnych osobników (nazw indywidualnych) a_1, \dots, a_n oraz dowolnych konceptów (atomowych lub złożonych) C i D , dowolnej roli R i dowolnej liczby naturalnej n poprawnie zbudowanymi konceptami formalizmu *SHOIQ* są wyrażenia:

A	<i>koncept atomowy</i>
$\{a_1, \dots, a_n\}$	<i>koncept wyliczony</i>
$\neg C$	<i>negacja konceptu</i>
$C \sqcap D$	<i>przecięcie konceptów</i>
$C \sqcup D$	<i>suma konceptów</i>
$\exists R.C$	<i>kwantyfikator egzystencjalny</i>
$\forall R.C$	<i>kwantyfikator ogólny</i>
$\geq n R.C$	<i>kwalifikowane dolne ograniczenie liczebnościowe</i>
$\leq n R.C$	<i>kwalifikowane górne ograniczenie liczebnościowe</i>

Oprócz wymienionych wyżej konstruktorów, w logice opisowej stosuje się także pojęcia *konceptu uniwersalnego* i *konceptu pustego*. Koncept uniwersalny (nazywany w anglojęzycznych publikacjach *top*) oznacza się znakiem \top i definiuje jako $\top = C \sqcup \neg C$. Koncept pusty (nazywany w anglojęzycznych publikacjach *bottom*) ma symbol \perp i jest definiowany jako $\perp = C \sqcap \neg C$.

Jak już wspomniano wcześniej, role odpowiadają predykatom binarnym. Role atomowe określane są nazwą, natomiast w formalizmie *SHOIQ* można dodatkowo sformułować tzw. *rolę odwrotną*:

Definicja 2.6 (konstruktory ról formalizmu SHOIQ)

Dla dowolnej nazwy (generalnej) R i dowolnej roli S poprawnie zbudowanymi rolami formalizmu *SHOIQ* są wyrażenia:

R	<i>rola atomowa</i>
S^-	<i>rola odwrotna</i>

■

Formalizmy logiki opisowej pozwalają także na formułowanie zdań. Zdania te tradycyjnie dzielą się dwie kategorie: aksjomaty i asercje.

Definicja 2.7 (aksjomaty i asercje formalizmu SHOIQ)

Dla dowolnych konceptów C i D oraz dowolnych ról R i S poprawnie zbudowanymi aksjomatami formalizmu *SHOIQ* są wyrażenia:

$C \sqsubseteq D$	<i>aksjomat subsumcji (podrzędności konceptów)</i>
$C \equiv D$	<i>aksjomat równoważności konceptów</i>
$R \sqsubseteq S$	<i>aksjomat podrzędności ról</i>
$\text{trans}(R)$	<i>aksjomat przechodniości ról</i>

Dla dowolnego konceptu C , roli R i osobników a i b poprawnie zbudowanymi asercjami formalizmu *SHOIQ* nazywamy wyrażenia:

$C(a)$	<i>asercja unarna</i>
$R(a, b)$	<i>asercja binarna</i>

■

Ontologia $O = (T, A)$ sformułowana w danym formalizmie logiki opisowej składa się z dwóch części: *terminologii*, czyli zbioru aksjomatów, tradycyjnie oznaczanego jako T , zwanego też czasem Tboxem, oraz z *opisu świata*, czyli zbioru asercji, tradycyjnie oznaczanego jako A , zwanego też czasem Aboxem lub *zbiorem faktów*. Aksjomaty zawarte w terminologii opisują zależności subsumcji między konceptami (atomowymi lub złożonymi). W formalizmach o wyższej ekspresywności, jak na przykład *SHOIQ*, pozwalają też opisać relację subsumcji między rolami. Gramatyka pozwala także opisać role poprzez wyrażenie ich dziedziny i przeciwdziedziny w postaci konstruktorów konceptów. Dziedzinę roli wyraża się aksjomatem $\exists R. \top \sqsubseteq C$. Koncept C ogranicza od góry dziedzinę roli R . Z kolei przeciwdziedzinę, zwaną w logice opisowej *zakresem*, opisuje się aksjomatem $\top \sqsubseteq \forall R. C$. Oznacza to, że w tym przypadku koncept C ogranicza od góry zakres roli.

Tworzące opis świata asercje wyrażają przynależność osobników, których nazwy indywidualne są znane, do ekstensji konceptów, lub przynależność par osobników do ekstensji ról.

Każda ontologia posiada swoją *sygnaturę*, zwaną też czasem słownikiem. Sygnatura ontologii O , oznaczana $\text{Sig}(O)$, jest sumą rozłączną nazw konceptów atomowych $\text{Sig}_C(O)$, ról atomowych $\text{Sig}_R(O)$ oraz osobników $\text{Sig}_A(O)$ wykorzystanych w ontologii. Sygnatury terminologii $\text{Sig}(T) = \text{Sig}_C(T) \uplus \text{Sig}_R(T) \uplus \text{Sig}_A(T)$ oraz opisu świata $\text{Sig}(A) = \text{Sig}_C(A) \uplus$

$Sig_R(A) \uplus Sig_A(A)$ jest definiowany analogicznie. Oczywiście dla $O = (T, A)$ zachodzi $Sig(O) = Sig(T) \cup Sig(A)$.

Zbiór wszystkich poprawnych aksjomatów, jakie można zbudować z wykorzystaniem danej sygnatury (słownika) S w danym formalizmie \mathcal{L} , oznaczamy jako $\mathcal{L}_T(S)$, zaś odpowiedni zbiór wszystkich poprawnych asercji jako $\mathcal{L}_A(S)$. Związki pomiędzy wprowadzonymi pojęciami formalizuje poniższa definicja.

Definicja 2.8 (ontologia dla formalizmu \mathcal{L})

Ontologia O o sygnaturze $S = Sig(O)$ wyrażona w formalizmie \mathcal{L} logiki opisowej jest parą (T, A) , gdzie $T \subset \mathcal{L}_T(S)$ jest skończonym zbiorem aksjomatów nazywanym *terminologią*, zaś $A \subset \mathcal{L}_A(S)$ jest skończonym zbiorem asercji nazywanym *opisem świata*. ■

Warto tutaj umieścić kilka słów wyjaśnienia. Autor niniejszej pracy używa nazwy *formalizm logiczny* w miejscu, gdzie w innych publikacjach mówi się o *językach logicznych*. Każdy język musi posiadać słownik (zwany też alfabetem [KM1979]), który zawiera symbole stanowiące podstawowy budulec zdań (słów nad alfabetem). Formalizm logiczny dysponuje tylko częścią słownika, gdyż nie zawiera symboli pozallogicznych. Nazwanie formalizmu logicznego językiem jest więc usprawiedliwione jedynie wówczas, gdy założy się, że posiada on jakiś nieskończony słownik, zawierający wszystkie słowa użyte we wszystkich zdaniach, które kiedykolwiek zostaną wypowiedziane w ramach tego formalizmu.

Podejście takie jest bardzo użyteczne w sytuacji, gdy abstrahuje się od problemów konceptualizacji, jednakże należy pamiętać, że stoi ono w sprzeczności z ważnymi dla niniejszej rozprawy rozważaniami N. Guarino, zaprezentowanymi w punkcie 2.1.4. Według tych rozważań każdy słownik posiada przyporządkowanie ontologiczne, a zadaniem ontologii jest jedynie jego opisanie. Jeśli więc dwie ontologie o równych sygnaturach (słownikach) i wyrażone zgodnie z tym samym formalizmem logicznym opisują różne konceptualizacje (inaczej definiują zależności między pojęciami), to opisują one przyporządkowania ontologiczne dwóch różnych języków o (przypadkowo) takich samych słownikach. Kierując się tym stanowiskiem, autor starał się w bieżącej pracy trzymać zasady, że słownik, a zatem i język, jest wyznaczany przez konceptualizację, a formalizm logiczny nadaje jedynie temu językowi gramatykę i semantykę.

Jednakże w niektórych sytuacjach użyteczne będzie chwilowe odejście od tej zasady i przyjęcie, za pracami źródłowymi, że formalizm \mathcal{L} jest językiem o nieskończonym słowniku. Na przykład w tym miejscu podejście to umożliwi dokonanie uzgodnienia, że symbolem \mathcal{L}_T będzie oznaczany zbiór wszystkich poprawnie zbudowanych aksjomatów tego formalizmu (dla nieskończonej sygnatury), a symbolem \mathcal{L}_A zbiór wszystkich poprawnie zbudowanych asercji.

Semantyka konstrukcji syntaktycznych logiki opisowej określona jest przez definicję interpretacji oraz modelu ontologii.

Definicja 2.9 (interpretacja dla formalizmu $SHOIQ$)

Interpretacją bazową \mathcal{I} dla sygnatury $S = S_C \uplus S_R \uplus S_A$ dla formalizmu $SHOIQ$ nazywa się parę $(\Delta^{\mathcal{I}}, \bullet^{\mathcal{I}})$, gdzie $\Delta^{\mathcal{I}}$ jest dowolnym niepustym zbiorem nazywanym *dziedziną interpretacji* \mathcal{I} , zaś $\bullet^{\mathcal{I}}$ jest *przyporządkowaniem interpretacyjnym* przypisującym każdemu elementowi sygnatury jego odniesienie przedmiotowe, czyli każdemu konceptowi atomowemu $A \in S_C$ podzbiór dziedziny $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, każdej roli atomowej $R \in S_R$ podzbiór par elementów dziedziny $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, zaś każdemu osobnikowi $a \in S_A$ element dziedziny $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Pojęcie interpretacji bazowej rozszerzane jest na *interpretację* poprzez przypisanie ekstensji konceptom i rolom złożonym (nieatomowym) w następujący sposób:

$$\begin{aligned}
\{a_1, \dots, a_n\}^I &= \{a_1^I, \dots, a_n^I\} \\
(\neg C)^I &= \Delta^I \setminus C^I \\
(C \sqcap D)^I &= C^I \cap D^I \\
(C \sqcup D)^I &= C^I \cup D^I \\
(\exists R.C)^I &= \{e \mid e \in \Delta^I \wedge \exists (e, e') \in R^I (e' \in C^I)\} \\
(\forall R.C)^I &= \{e \mid e \in \Delta^I \wedge \forall (e, e') \in R^I (e' \in C^I)\} \\
(\geq n R.C)^I &= \{e \mid e \in \Delta^I \wedge \|\{(e, e') \mid (e, e') \in R^I \wedge (e' \in C^I)\}\| \geq n\} \\
(\leq n R.C)^I &= \{e \mid e \in \Delta^I \wedge \|\{(e, e') \mid (e, e') \in R^I \wedge (e' \in C^I)\}\| \leq n\} \\
(R^-)^I &= \{(e, e') \mid (e', e) \in R^I\} \\
\top^I &= \Delta^I \\
\perp^I &= \emptyset
\end{aligned}$$

■

Zbiór wszystkich interpretacji dla danej sygnatury \mathcal{S} jest oznaczany jako $\mathcal{I}^\sigma(\mathcal{S})$.

Aby interpretacja była modelem ontologii O , musi *spełniać* wszystkie zdania (aksjomaty i asercje) zawarte w O . Pojęcie modelu zdefiniowano poniżej.

Definicja 2.10 (spełnianie i model SHOIQ)

Interpretacja \mathcal{I} dla sygnatury $\mathcal{S} = \text{Sig}(O)$ jest modelem danej ontologii $O = (T, A)$ (co zapisuje się $\mathcal{I} \models O$; można także powiedzieć, że \mathcal{I} spełnia O), jeżeli spełnia terminologię T ($\mathcal{I} \models T$) i opis świata A ($\mathcal{I} \models A$). Interpretacja spełnia terminologię T jeżeli spełnia każdy aksjomat α w niej zawarty (fakt spełniania aksjomatu α zapisuje się jako $\mathcal{I} \models \alpha$). Warunki spełniania poszczególnych typów aksjomatów zebrane są poniżej:

$$\mathcal{I} \models C \sqsubseteq D \iff C^I \subseteq D^I$$

$$\mathcal{I} \models C \equiv D \iff C^I = D^I$$

$$\mathcal{I} \models R \sqsubseteq S \iff R^I \subseteq S^I$$

$$\mathcal{I} \models \text{trans}(R) \iff \forall e, e', e'' ((e, e') \in R^I \wedge (e', e'') \in R^I \rightarrow (e, e'') \in R^I)$$

Interpretacja spełnia opis świata A jeżeli spełnia każdą asercję σ w niej zawartą (fakt spełniania asercji σ zapisuje się jako $\mathcal{I} \models \sigma$). Warunki spełniania poszczególnych typów asercji zebrane są poniżej:

$$\mathcal{I} \models C(a) \iff a^I \in C^I$$

$$\mathcal{I} \models R(a, b) \iff (a^I, b^I) \in R^I$$

■

Zamieszczony w niniejszym punkcie opis logiki opisowej warto uzupełnić krótkim przeglądem innych niż SHOIQ formalizmów. Poszczególne litery w akronimie oznaczają możliwość zastosowania pewnych określonych konstrukcji, co oznacza, że jeśli niektóre z nich zostaną usunięte, to akronim będzie oznaczał formalizm prostszy. I tak w SHIQ nie można używać konceptów wyliczanych, w SHQ dodatkowo ról odwrotnych, w SH ograniczeń liczebnościowych, zaś w S aksjomatów podrzędności ról. Samo S jest z kolei skrótem dla \mathcal{ALCR}^+ , przy czym w formalizmie \mathcal{ALC} nie można używać aksjomatu przechodniości roli. Akronimy innych formalizmów można budować na analogicznej zasadzie, np. formalizm \mathcal{ALCH} dozwalałby na użycie tych samych konstrukcji co \mathcal{ALC} z dodatkową możliwością użycia aksjomatu podrzędności ról. W pracach stosunkowo rzadko rozważa się formalizmy wykraczające ekspresywnością poza SHOIQ, wyjątkiem jest SROIQ, w którym \mathcal{R} zastępujący \mathcal{H} oznacza możliwość wykorzystania kilku specyficznych konstrukcji dla ról (m.in. aksjomatu zwrotności i symetrii), formalizmy z \mathcal{B} (zastępującym \mathcal{C}) w akronimie, w których możliwe jest wykorzystanie przecięcia, sumy i negacji ról (z reguły

są to formalizmy takie jak \mathcal{ALB} lub o podobnej ekspresywności, gdyż dodanie przecięcia, sumy i negacji roli do bardziej ekspresywnych formalizmów spowodowałoby utratę ich rozstrzygalności), oraz formalizmy umożliwiające wykorzystanie tzw. *dziedzin konkretnych*.

Dziedziny konkretne stanowią bardzo ważne rozszerzenie logiki opisowej. Choć logikę opisową przyrównuje się często do metod obiektowych (patrz np. [BL2004]), a ich pochodzenie wywodzi się od ramek Minsky’ego, formalizmy DL nie zawierają środków wyrażania wartości tekstowych i liczbowych skojarzonych z poszczególnymi osobnikami. Innymi słowy logika opisowa w podstawowej wersji nie zawiera mechanizmów obsługi *atrybutów* (zawierających wartości symboliczne).

Brak ten został dosyć szybko zidentyfikowany i wskazany przez środowiska zajmujące się rozwojem tej dziedziny wiedzy. Aby go uzupełnić, utworzono pojęcie tzw. dziedzin konkretnych. Wykorzystanie dziedziny konkretnej \mathcal{D} powodowało rozszerzenie dziedziny interpretacji o zbiór $\Delta^{\mathcal{D}}$ określający możliwe wartości atrybutów \mathcal{D} . *Atrybut* z kolei oznacza specjalny rodzaj roli, takiej, której odniesienie przedmiotowe może łączyć ze sobą jedynie elementy dziedziny podstawowej (abstrakcyjnej) Δ z elementami zbioru $\Delta^{\mathcal{D}}$. Przykładami dziedzin konkretnych są:

- \mathcal{N} , czyli dziedzina wszystkich liczb naturalnych,
- \mathcal{R} , liczby rzeczywiste,
- \mathcal{DB} , dziedzina obejmująca wartości atomowe możliwe do wykorzystania w relacyjnej bazie danych.

Dla każdej dziedziny konkretnej definiowany jest pewien zbiór predykatów $pred(\mathcal{D})$. Z każdym predykatem P z tego zbioru skojarzona jest jego krotność $n(P)$. Predykat P jest spełniony dla swojej ekstensji $P^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^{n(P)}$. Ponadto zbiór predykatów powinien być tak zbudowany, aby z każdym predykatem P skojarzyć można było także predykat $Q \in pred(\mathcal{D})$, taki że $Q^{\mathcal{D}} = (\Delta^{\mathcal{D}})^{n(P)} \setminus P^{\mathcal{D}}$.

Wykorzystanie predykatów pozwala na zastosowanie specjalnego konstruktora konceptów o postaci $\exists(u_1, u_2, \dots, u_n).P$, gdzie dla każdego $i \in [1, n]$ u_i jest atrybutem, zaś P jest n -arnym predykatem z $pred(\mathcal{D})$. Ekstensją takiego konceptu są osobniki e , dla których wartości atrybutów u_1, u_2, \dots, u_n spełniają predykat P (co formalnie można wyrazić jako zbiór $\{e \in \Delta \mid \exists \{v_i\}_{i \in [1, n]} (\forall j \in [1, n]) (e, v_j) \in u_j^{\mathcal{I}} \wedge (v_1, v_2, \dots, v_n) \in P^{\mathcal{D}}\}$).

Dziedziny konkretne nie są oczywiście jedynym przydatnym rozszerzeniem formalizmu logiki opisowej. Zostały one przedstawione w tym punkcie, gdyż będą wykorzystane w niektórych przykładach przedstawionych w niniejszej pracy.

2.2.3 Systemy wnioskujące i języki

Wnioskowanie z ontologii wykorzystujących logikę opisową odbywa się zazwyczaj za pomocą algorytmów *tableau*. Algorytmy te orzekają w zasadzie o spełnialności ontologii, a sprawdzają to poprzez systematyczne próby zbudowania przynajmniej jednego jej modelu. Model budowany jest w sposób pośredni, za pomocą rozbudowy tzw. *tableau* mającego postać grafu. Węzły tego grafu reprezentują osobniki, zaś krawędzie wystąpienia ról pomiędzy osobnikami. Każdy węzeł i każda krawędź zawiera zestaw etykiet, które dla węzłów wskazują, do których (ekstensji) konceptów należy dany osobnik, zaś dla krawędzi, wystąpienie których ról reprezentuje dana krawędź.

Budowa *tableau* dla ontologii zawierających jedynie terminologię T (czyli pusty Abox A) rozpoczyna się od pojedynczego węzła (*osobnik inicjalny*). Temu pojedynczemu węzłowi

przypisywana jest etykieta reprezentująca tzw. koncept zinternalizowany. Koncept ten powstaje w dwóch krokach. Na początek tworzony jest koncept C' zbierający informację ze wszystkich aksjomatów podrzędności i równoważności:

$$C' = \bigwedge_{C \sqsubseteq D \in T} (\neg C \sqcup D) \sqcap \bigwedge_{C \equiv D \in T} ((\neg C \sqcup D) \sqcap (C \sqcup \neg D)) \quad (2.1)$$

Koncept ten budowany jest na zasadzie wykluczenia: aksjomat $C \sqsubseteq D$ wyklucza (w spójnej bazie wiedzy) przynależność osobnika do konceptu $(C \sqcap \neg D)$ — co, odwracając zdanie, oznacza, że każdy osobnik musi przynależeć do konceptu $\neg(C \sqcap \neg D) \equiv \neg C \sqcup D$. Aksjomat równoważności $C \equiv D$ jest z kolei traktowany jako dwa aksjomaty $C \sqsubseteq D$ i $D \sqsubseteq C$.

Koncept zinternalizowany C'' powstaje jako przecięcie konceptu C' z konceptem $\forall U.C''$, gdzie U to specjalna, sztucznie wprowadzona rola, od której dziedziczą wszystkie role wykorzystane w ontologii. Wprowadzenie takiej konstrukcji gwarantuje, że każdy osobnik połączony jakąkolwiek rolą z osobnikiem inicjalnym również będzie musiał przynależeć do konceptu C' oraz C'' .

Po tych wstępnych krokach *tableau* rozbudowywane jest zgodnie ze zdefiniowanymi dla danego formalizmu logiki opisowej regułami. Reguły te stosują się do osobników oznaczonych odpowiednimi etykietami i ściśle odpowiadają wykorzystywanym konstruktorom konceptów. Na przykład reguła „ \sqcap ” mówi, że jeżeli w *tableau* istnieje węzeł (osobnik) z etykietą „ $C \sqcap D$ ” i nie ma on etykiet „ C ” i „ D ” (gdzie C i D to dowolne — być może złożone — koncepty), to należy jego zbiór etykiet o „ C ” i „ D ” uzupełnić.

Z punktu widzenia algorytmu *tableau* szczególną uwagę trzeba zwrócić na wykonywanie dwóch rodzajów reguł. Pierwszy to reguły niedeterministyczne, których przykładem jest reguła „ \sqcup ”. Stanowi ona, że węzłowi grafu (osobnikowi) z etykietą „ $C \sqcup D$ ” należy przypisać etykietę „ C ” lub „ D ”. W praktyce i w pewnym uproszczeniu oznacza to, że dalszą część algorytmu należy wykonać dwa razy dla obu tych możliwości. Drugim rodzajem reguł są reguły produkujące osobniki, których przykładem jest reguła „ \exists ”. Reguła ta stanowi, że dla węzła opatrzonego etykietą „ $\exists R.C$ ” nie połączonego krawędzią zaetykietowaną „ R ” z węzłem, w którego zbiorze etykiet znajduje się „ C ”, należy taką krawędź i taki węzeł utworzyć. Reguły te powodują zatem rozbudowę grafu *tableau*.

Z regułami produkującymi osobniki wiąże się ważne zagadnienie zapewnienia, aby algorytm *tableau* zakończył swoją pracę. By osobników nie produkować w nieskończoność (co byłoby możliwe, gdyby na przykład istniał w danej ontologii aksjomat $\top \sqsubseteq \exists maKrewnego.\top$ oznaczający, że każdy osobnik musi mieć krewnego), wprowadza się tzw. *strategie blokowania*. Strategie te przerywają proces rozbudowy *tableau*, kiedy stwierdzą, że jeden z już utworzonych osobników można wykorzystać jako węzeł, do którego prowadzić będzie nowo utworzona krawędź.

Algorytmy *tableau* stanowią zatem poprawną i pełną (tj. odnalezienie *tableau* jest gwarancją istnienia modelu i, odwrotnie, nieodnalezienie *tableau* jest gwarancją nieistnienia modelu) metodę wnioskowania o spełnialności ontologii. Mogą zostać także zaadaptowane do rozstrzygania o innych problemach wnioskowania z terminologii (np. sprawdzenie, czy $C \sqsubseteq D$, może być przeprowadzone przez dodanie do inicjalnego osobnika etykiety „ $C \sqcap \neg D$ ”), a także do wnioskowania z opisu świata (wówczas inicjalna postać *tableau* staje się bardziej skomplikowana, gdyż musi zawierać wszystkie informacje wyrażone w opisie świata).

Rozwój inicjatywy Semantic Web spowodował wzrost popularności programów wnioskujących z ontologii w logice opisowej. Najbardziej popularnymi przykładami tych aplikacji są obecnie FaCT [H1998], Pellet [SPGKK2007], Racer [HM2001] oraz KAON2 [HMS2004]. Trzy pierwsze posługują się najbardziej rozpowszechnioną techniką

wnioskowania opartą na wykorzystaniu opisanych wyżej algorytmów *tableau*. Z kolei KAON2 przekształca ontologie zapisane w logice opisowej na reguły i wnioskuje z nich przy użyciu technik właściwych dla tej formy reprezentacji wiedzy.

Choć wymienione systemy charakteryzują się dobrą wydajnością i zawierają wiele optymalizacji pozwalających im w różnych sytuacjach na rozwiązywanie problemów wnioskowania (często o złożoności ponadwykładniczej) w akceptowalnym czasie, przetwarzanie bardzo dużych ontologii często leży poza ich zasięgiem. Na przykład ontologii Galen [RR1996] nie udaje się wczytać większości systemów (patrz np. [MSH2007]). Dodatkowo wydajność systemów wnioskujących drastycznie spada wraz z liczbą osobników zawartych w opisie świata (patrz [W2007]).

W ramach inicjatywy Semantic Web do zapisu ontologii wykorzystuje się język OWL, który można traktować jako syntaktyczny wariant jednego z formalizmów logiki opisowej. W najnowszej wersji specyfikacji OWL pochodzącej z października 2009 (<http://www.w3.org/TR/owl2-overview/>) podstawowa wersja języka (o tzw. bezpośredniej semantyce, ang. *Direct semantics*) obejmuje swoją ekspresywnością formalizm *SROIQ* z możliwością obsługi dziedzin konkretnych (w poprzedniej wersji tzw. podjęzyk — ang. *sublanguage* — OWL-DL obejmował swoją ekspresywnością podzbiór formalizmu *SHOIQ*).

Podstawowa składnia OWL oparta jest na RDF, czyli zakłada, że dokument zbudowany jest z trójek. Specyfikacja określa zasady kodowania odpowiednich asercji i aksjomatów przy użyciu tych uproszczonych zdań. Trójki RDF można zapisywać w XML-u i większość ontologii w Internecie dostępnych jest właśnie w postaci plików XML.

Remedium na problemy wnioskowania z dużych ontologii mogłaby być ich modularyzacja. Język OWL sam z siebie daje jednak bardzo niewielkie możliwości, jeśli chodzi o wydzielanie i łączenie modułów. Jediną dostępną konstrukcją jest tu specjalne polecenie *import* powodujące załączenie w całości zawartości innego pliku.

2.3 Podsumowanie

W niniejszym rozdziale przedstawiono szereg zagadnień dotyczących wykorzystania ontologii w informatyce. Opisano filozoficzne znaczenie tego słowa oraz znaczenie węższe, traktujące ontologię jako teorię aksjomatyczną. Zaprezentowano szereg formalizmów logicznych używanych do wyrażania ontologii oraz związane z nimi systemy wnioskujące. Opisano także rolę ontologii widzianej jako fragment szerszego procesu konceptualizacji. Rozdział przedstawia też sposoby wykorzystania ontologii w informatyce, począwszy od systemów regułowych opartych na różnych odmianach algorytmu rezolucji, skończywszy na zastosowaniu ontologii w ramach inicjatywy Semantic Web.

Inicjatywa Semantic Web, a przede wszystkim ustalenie standardu w postaci języka OWL, spowodowała wzrost zainteresowania ontologicznym opisem różnych dziedzin i powstanie wielu ontologii dostępnych w Internecie. Wkrótce jednak okazało się, że mają one szereg zasadniczych wad w znacznym stopniu ograniczających możliwość ich praktycznego zastosowania. We wstępie do niniejszej pracy wymieniono pięć problemów referencyjnych, które ograniczają stosowalność ontologii w praktyce. Najbardziej dotkliwym spośród nich jest problem integracyjny, czyli trudność, czasem wręcz niemożność powtórnego wykorzystania. Kiedy tworzona jest nowa ontologia opisująca jakąś dziedzinę, sensowne byłoby nie definiować wszystkich pojęć od początku, lecz skorzystać w części z istniejących już w sieci globalnej różnych innych ontologii z tej lub pokrewnych dziedzin. Dla przykładu, tworząc system wnioskujący mający wspomóc działanie jakiejś firmy, dobrze byłoby skorzystać z ontologicznych opisów powstałych dla potrzeb podobnych firm, dla potrzeb opisu rynku, na

którym ta firma działa, lub dla potrzeb producentów dostarczających firmie towary lub usługi. Okazuje się jednak, że powstałe niezależnie ontologie różnią się od siebie w znacznym stopniu sposobem, w jaki została przeprowadzona konceptualizacja. Wyodrębnione zostały różne relacje konceptualne. Te bardziej podobne różnią się stopniem lub nazwą. Nazwy relacji są mylące — często podobna relacja ma inną nazwę, kiedy indziej podobna nazwa oznacza inną relację. Najgorsze jest jednak to, że powtórne wykorzystanie istniejącej już ontologii działa na zasadzie „wszystko albo nic”, nie istnieją bowiem sprawne techniki pozwalające wyodrębnić tylko fragment danej ontologii. Tak więc jedynym rozwiązaniem jest w tej sytuacji stworzenie własnej ontologii opisującej całą dziedzinę i wszystkie aspekty funkcjonowania tworzonego systemu. Wynikiem są duże i skomplikowane ontologie, co w znacznym stopniu utrudnia proces ich wytwarzania. Ponieważ nie ma możliwości podzielenia ich na moduły, bardzo trudno jest poprawnie zorganizować pracę zespołów projektujących i implementujących, a także trudno jest utrzymać kontrolę nad treścią aksjomatów. Problemem staje się wewnętrzne uzgodnienie konceptualizacji. Prowadzi to do niekoherencji, niespójności⁹ i do dublowania definicji różnych pojęć.

Trudność ponownego użycia i wszystkie jej konsekwencje zdeterminowały kierunki prac zmierzających do udoskonalenia ontologii. Wśród nich da się wyróżnić dwa główne: normalizacja i modularyzacja baz wiedzy. Pierwszy kierunek to rozwiązania, których celem jest spowodowanie, aby tworzone ontologie były podobne. Badacze, którzy je proponują wychodzą z założenia, że istnieje jeden możliwy opis świata, który w swej pełnej postaci jest nikomu nieznanym, a ontologie tworzone przez różnych twórców są jego fragmentami. Fragmenty te, jako że są fragmentami tej samej całości, gdy dotyczą tego samego lub podobnego zagadnienia, muszą być podobne. Zagadnienia związane z normalizacją ontologii są opisane w rozdziale 3. Drugi kierunek proponowanych rozwiązań, modularyzacja, wychodzi z zupełnie odwrotnego założenia. Nie istnieje jeden opis świata, wszystko zależy od kontekstu. Od baz wiedzy nie można żądać, aby ujmowały problemy w podobny sposób, natomiast rozwiązań należy szukać w sposobie wykorzystania rozproszonej w nich wiedzy. Problemy kontekstowego podejścia do opisu dziedzin oraz propozycje różnych technik modularyzacji baz wiedzy są opisane w rozdziale 4.

⁹ Niekoherencją nazywamy sytuację, gdy w terminologii istnieją niespełnialne koncepty, z kolei niespójność jest przypisaniem danego osobnika do konceptów, które są rozłączne.

3 Normalizacja ontologii

Jednym z kierunków, w których podążają rozwiązania mające za cel złagodzenie opisanych we wstępie do niniejszej rozprawy problemów referencyjnych, w szczególności do rozwiązania problemów konstrukcyjnego i integracyjnego, jest dążenie do narzucenia twórcom ontologii pewnych standardów, według których powinni postępować, dokonując konceptualizacji danej dziedziny. Dadzą się tu wyodrębnić dwie metody. Jedna polega na opracowaniu systemu norm, które spowodują, że podobne problemy będą rozwiązywane w podobny sposób. Najbardziej znanym zbiorem zasad postępowania dla twórców baz wiedzy jest metoda walidacji ontologii OntoClean. Jest ona opisana w podrozdziale 3.1. Proponuje ona szereg kryteriów, które należy wziąć pod uwagę podczas tworzenia relacji taksonomicznych pomiędzy encjami. Jest więc podstawą dla stworzenia poprawnego modelu opisywanej dziedziny.

Druga metoda normalizacji ontologii polega na wykorzystaniu podczas tworzenia tzw. *ontologii dziedzinowych*, czyli ontologii opisujących jakąś dziedzinę szczegółową, gotowych ontologii wysokiego poziomu (ang. *top-level ontologies*) w sposób abstrakcyjny opisujących podstawowe zjawiska ze świata rzeczywistego. Narzucenie wszystkim ontologiom dziedzinowym wspólnego opisu zjawisk podstawowych ma sprawić, że będą one w podobny sposób ujmowały zjawiska znajdujące się na niższym poziomie abstrakcji. Zagadnienie wykorzystania ontologii wysokiego poziomu jest opisane w podrozdziale 3.2.

Podrozdział 3.3 podsumowuje niniejszy rozdział, zawiera również ocenę metod normalizacji w aspekcie ich przydatności do rozwiązania problemów ograniczających praktyczne zastosowanie ontologii.

3.1 Metoda walidacji ontologii OntoClean

Metoda OntoClean została zaproponowana w szeregu artykułów w latach 2000-2001 przez N. Guarino i C. A. Welty'ego ([GW2000a], [GW2000b], [GW2000c]). Pozytywne, niekiedy entuzjastyczne opinie na temat tej metody (wg [GW2004]) padły z ust m. in. znanego praktyka w dziedzinie tworzenia ontologii A. Rectora oraz twórców ontologii CYC.

Autorzy zmieniają nieco słownik znany z inżynierii oprogramowania lub z prac nad sztuczną inteligencją, redefiniując słowa takie jak *encja* i *właściwość* zgodnie z ich pierwotnym, wziętym z filozofii bytu znaczeniem. Tak więc w dalszej części rozdziału encja będzie oznaczała obiekt dziedziny, a właściwość — przynależność encji do ekstensji jakiegoś pojęcia. Na przykład *bycie jabłkiem*, *bycie stołem* lub *bycie czerwonym* jest właściwością encji. Nie należy tak rozumianej właściwości utożsamiać ani z pojęciem klasy, ani z pojęciem predykatu. Ma ona charakter intensjonalny, więc reprezentuje *znaczenie* wyrażenia. Klasa jest ekstensjonalną reprezentacją właściwości, jest zbiorem encji (obiektów) będących wystąpieniami tej klasy, encji *wykazujących* tę właściwość. Z kolei predykat jest tworem syntaktycznym, symbolem reprezentującym klasę.

W punkcie 3.1.1 przedstawione są podstawy filozoficzne metody OntoClean oraz wymienione są podstawowe metawłaściwości. Punkt 3.1.2 formułuje główne reguły normalizujące proces konceptualizacji. W punkcie 3.1.3 przedstawiono szereg dodatkowych metawłaściwości wzbogacających charakterystykę właściwości użytych do opisu danej dziedziny. Punkt 3.1.4 zawiera dwa przykłady prezentujące możliwości metody. Zadaniem punktu 3.1.5 jest pokazanie, że metoda OntoClean jest metodą ogólną, która może być stosowana nie tylko w procesie tworzenia ontologii, lecz także wszędzie tam, gdzie projektant dokonuje konceptualizacji. Pokazano to na przykładzie metody relacyjnej.

3.1.1 Metawłaściwości w procesie konceptualizacji

Guarino i Welty opierają swoją metodę na *metawłaściwościach*, czyli na właściwościach właściwości. Głównym podziałem, którego dokonują, jest podział na właściwości *immanentne* (*rigid, essential*), właściwości *przenoszące tożsamość* (*carrying identity*) lub jej *dostarczające* (*supplying identity*), właściwości mówiące, że dana encja jest *całością* (*carrying unity*) złożoną z części, oraz właściwości *zależne* (*dependent*). W kolejnych punktach omówione zostanie znaczenie tych metawłaściwości.

3.1.1.1 Immanentność

To, że właściwość jest immanentna, oznacza, że posiadanie jej jest dla danej encji konieczne bez względu na przyjętą konceptualizację. W ramach metody OntoClean zdefiniowano immanentność, nieimmanentność (*non-rigidity*) i przeciwiimmanentność (*anti-rigidity*) za pomocą operatora modalnego:

Definicja 3.1 (właściwość immanentna, nieimmanentna i przeciwiimmanentna)

Właściwość φ jest *immanentna* wtwg $\forall x (\varphi(x) \rightarrow \Box \varphi(x))$.

Właściwość φ jest *nieimmanentna* wtwg $\exists x (\varphi(x) \wedge \neg \Box \varphi(x))$.

Właściwość φ jest *przeciwiimmanentna* wtwg $\forall x (\varphi(x) \rightarrow \neg \Box \varphi(x))$. ■

Właściwość jest immanentna, jeśli dla każdej encji, która ją wykazuje, jest ona *koniecznie* (czyli we wszystkich światach możliwych) jej właściwością. Odnosząc to do danej ontologii, właściwość jest immanentna, jeśli twórcy tej ontologii wykluczają jakąkolwiek możliwość takiego jej rozwoju, który pewnym encjom odebrałby tę właściwość. Taką właściwością jest na przykład *bycie człowiekiem*. Jeśli dana encja wykazuje taką właściwość (jeśli dany obiekt jest człowiekiem), to będzie ją wykazywała zawsze (to musi nim być zawsze).

Właściwość jest nieimmanentna, jeżeli istnieje taka encja wykazująca tę właściwość, dla której nie jest ona koniecznie jej właściwością. Po prostu gdy istnieje jakakolwiek wątpliwość, że właściwość jest immanentna, to jest ona nieimmanentna. Przykładem może być tutaj właściwość *bycia arcydziełem literackim*. Najczęściej arcydzieło literackie pozostaje arcydziełem na zawsze, a upływający czas jedynie potwierdza jego znaczenie. Może się jednak zdarzyć, że jakieś dzieło literackie, mimo że początkowo wysoko cenione, przestaje być takim po upływie jakiegoś czasu.

Właściwość jest przeciwiimmanentna, jeśli dla każdej encji wykazującej tę właściwość nie jest ona koniecznie jej właściwością. W przypadku tej właściwości istnieje pewność, że każda encja ją wykazująca w jakichś warunkach może ją stracić. Przykładem może być właściwość *bycia studentem*. Wiadomo, że każdy student jest nim jedynie w pewnym okresie czasu.

W trakcie dokonywania oceny jakości ontologii właściwości immanentne otrzymują oznaczenie $+R$, właściwości nieimmanentne $-R$, a właściwości przeciwiimmanentne $\sim R$. Zapisuje się więc: Człowiek $^{+R}$, Student $^{-R}$. Warto też zauważyć, że immanentność nie jest metawłaściwością dziedziczną. Aksjomat Student \sqsubseteq Człowiek nie sprawia, że właściwość *bycia studentem* staje się immanentna.

3.1.1.2 Tożsamość

Tożsamość (*identity*) jest metawłaściwością, która jest związana ze zdolnością ludzkiego umysłu do wyodrębniania encji (obiektów) w obrębie obserwowanego świata. Przy czym rozciąga się ona nie tylko na obiekty fizyczne, ale także na obiekty abstrakcyjne. Według [GW2000c] w opracowaniach filozoficznych mówi się o *kryterium tożsamości*

(*identity condition*) dla danej właściwości φ , która jest definiowana jako relacja ρ spełniająca następującą formułę:

$$\varphi(x) \wedge \varphi(y) \rightarrow (\rho(x, y) \leftrightarrow x = y) \quad (3.1)$$

Jeśli na przykład właściwością φ jest *bycie człowiekiem*, to relacja ρ o znaczeniu *mają taki sam odcisk palca* spełnia formułę (3.1) i może być zastosowana jako kryterium tożsamości. Jak widać, kryterium tożsamości jest warunkiem stwierdzającym równość (identyczność) encji oraz wynika z identyczności (między kryterium tożsamości a wyrażeniem stwierdzającym identyczność istnieje równoważność).

W praktyce dobór kryteriów tożsamości nie jest ani prosty, ani oczywisty. Choć jest to czynność arbitralna, to pociąga za sobą konsekwencje, których projektanci baz wiedzy powinni być świadomi. Przykład 3.1 przedstawiony w rozdziale 3.1.4 pokazuje, jak istotne jest właściwe określenie kryterium tożsamości dla decyzji projektowych dotyczących subsumcji.

Innym problemem związanym z określeniem kryterium tożsamości jest stwierdzenie, czy dana właściwość *dostarcza tożsamość*, czy tylko ją *przenosi*. I tak właściwość *bycia człowiekiem* dostarcza tożsamość, natomiast właściwość *bycia studentem* jedynie ją przenosi. Ma to związek z tym, że *bycie człowiekiem* jest immanentne, natomiast studentem można być (w wielu szkołach naraz) lub nie być.

Kolejnym problemem jest odpowiedź na pytanie, czy tożsamość dostarczana przez daną właściwość jest *synchroniczna* czy *diachroniczna*. Ta pierwsza jest związana z jakimś odcinkiem czasu, ta druga jest niezależna od rozpatrywanego momentu. Czy statek po przebudowie (np. polski transatlantyk „Stefan Batory” po przebudowie z holenderskiego „Maasdam”) jest nadal tym samym, czy innym statkiem? Czy człowiek po utracie świadomości jest tym samym człowiekiem (czy tożsamość jest określona przez jedność świadomości, czy przez cielesność)? Odpowiedź na te pytania zależy od przeznaczenia tworzonej bazy wiedzy.

Przez wzgląd na ten ostatni problem (i opowiadając się za tożsamością diachroniczną) autorzy metody OntoClean proponują definicję kryterium tożsamości, która rozszerza formułę (3.1), biorąc pod uwagę również czynnik czasu.

Definicja 3.2 (kryterium tożsamości)

Kryterium tożsamości właściwości φ jest formuła γ , która spełnia jedną z formuł:

$$E(x, t) \wedge \varphi(x, t) \wedge E(y, t') \wedge \varphi(y, t') \wedge x = y \rightarrow \gamma(x, y, t, t') \quad (3.2)$$

$$E(x, t) \wedge \varphi(x, t) \wedge E(y, t') \wedge \varphi(y, t') \wedge \gamma(x, y, t, t') \rightarrow x = y \quad (3.3)$$

Kryterium tożsamości jest konieczne, jeśli spełnia (3.2), a wystarczające, jeśli spełnia (3.3). Predykat $E(x, t)$ oznacza, że encja x jest *rzeczywista* i *istnieje* w chwili t (w sensie wy tłumaczonym w [GW2000b]). ■

Na bazie kryterium tożsamości zdefiniowane są kolejne metawłaściwości.

Definicja 3.3 (właściwość przenosząca kryterium tożsamości)

Właściwość φ *przenosi kryterium tożsamości* wtwg jest subsumowana przez właściwość dostarczającą to kryterium (włączając przypadek, gdy sama dostarcza to kryterium). ■

Definicja 3.4 (właściwość dostarczająca kryterium tożsamości)

Właściwość φ *dostarcza kryterium tożsamości* wtwg:

- 1) jest immanentna,
- 2) istnieje dla niej konieczne lub wystarczające kryterium tożsamości,

- 3) to samo kryterium tożsamości nie jest przenoszone przez wszystkie właściwości, które subsumują φ . Oznacza to, że jeżeli φ dziedziczy różne (choć kompatybilne) kryteria tożsamości od wielu różnych właściwości, to może być wciąż traktowana jako dostarczająca tożsamość. ■

Definicja 3.5 (sortal)

Każda właściwość, która przenosi kryterium tożsamości, jest nazywana *sortalem*. ■

Właściwości przenoszące kryterium tożsamości oznaczane są symbolem **+I**, te które go nie przenoszą symbolem **-I**. Właściwości dostarczające kryterium tożsamości oznaczane są symbolem **+O** (**-O** w przeciwnym przypadku). Symbol **+O** wymusza **+I** i **+R**.

3.1.1.3 Całość

Bycie całością jest zjawiskiem, które pociąga za sobą istnienie części. Autorzy metody OntoClean posilkują się definicją całości sformułowaną przez P. Simonsa (za [GW2000a]):

Każdy uczestnik podziału danego obiektu pozostaje w pewnej relacji do innych uczestników tego podziału; żaden uczestnik podziału nie jest w tej relacji z czymkolwiek innym niż uczestnicy tego podziału¹⁰.

Inaczej mówiąc, encje będące całością złożone są z części w taki sposób, że każda z tych części jest połączona z pozostałymi częściami tej całości i z niczym więcej. Bardziej formalnie, używając języka teorii mnogości, można powiedzieć, że podział obiektu tworzy klasę części, która całkowicie go wyczerpuje. Relacja, którą te części spełniają, musi być relacją równoważności.

Definicja 3.6 (całość)

Obiekt x jest *całością* ze względu na relację ω wtwg ω jest relacją równoważności taką, że wszystkie części obiektu x są połączone przez ω i ω nie łączy nic innego. ■

Aby właściwość mogła przenosić cechę bycia całością, musi być związana z taką relacją.

Definicja 3.7 (właściwość przenosząca całość)

Właściwość φ przenosi całość wtwg istnieje pojedyncza relacja równoważności ω taka, że każda encja wykazująca właściwość φ jest całością ze względu na ω . ■

Istnieje wiele przyczyn, które powodują, że umysł nadaje obserwowanym obiektom cechę całości. Są one badane zarówno przez filozofów, jak i psychologów. Wyodrębnia się (za [GW2000c]) trzy główne rodzaje relacji całości dla obiektów „konkretnych” (posiadających przestrzenno-czasową lokalizację):

- *Całość topologiczna* (ang. *topological unity*): bazująca na fizycznym połączeniu w całość rozumianą topologicznie, na przykład bryła węgla, kamień, owoc.
- *Całość morfologiczna* (ang. *morphological unity*): bazująca na kształcie (na przykład kulistym — piłka), ale niekoniecznie stanowiąca całość topologiczną (na przykład konstelacja gwiazd).
- *Całość funkcjonalna* (ang. *functional unity*): bazująca na przeznaczeniu obiektu, na przykład komplet sztućców, bikini.

¹⁰ W oryginale: *Every member of some division of the object stands in a certain relation to every other member, and no member bears this relation to anything other than members of the division.*

Części, które składają się na całość, mogą być również całościami. Jednak nie zawsze podlegają one tej samej relacji, względem której stanowią całość. Najczęściej relacja ta nie jest przechodnia.

Właściwości, które przenoszą kryterium całości, oznaczane są symbolem +U. Niektóre właściwości nie przenoszą jednolitego kryterium całości dla wszystkich encji, które je wykazują. Na przykład właściwość *Podatnik* obejmuje zarówno ludzi, jak i przedsiębiorstwa. Są one oznaczane symbolem –U.

Autorzy metody OntoClean wyodrębniają jeszcze grupę właściwości posiadające cechę *przeciwności*.

Definicja 3.8 (przeciwność)

Właściwość ma metawłaściwość *przeciwności* wtmg każda encja wykazująca tę właściwość nie jest całością. ■

Przykładem są właściwości obejmujące rzeczy niepoliczalne: *bycie wodą*, *bycie materią*. Właściwości te oznaczane są symbolem ~U.

3.1.1.4 Zależność

Zależność dotyczy ontologicznych związków między właściwościami. Autorzy metody OntoClean bazują na pojęciu *zależności pojęciowej* Simonsa [S1987].

Definicja 3.9 (właściwość zewnętrznie zależna)

Właściwość φ jest *zewnętrznie zależna* od właściwości ψ , jeżeli dla każdej encji x wykazującej właściwość φ *koniecznie* istnieje jakaś encja y wykazująca właściwość ψ , która nie jest ani częścią, ani konstytuentem x (patrz punkt 3.1.3 — właściwości mereologicznie ekstensjonalne). Można formalnie zapisać, że φ musi spełniać formułę:

$$\forall x \Box (\varphi(x) \rightarrow \exists y \psi(y) \wedge \neg P(x, y) \wedge \neg C(x, y))$$

gdzie $P(x, y)$ jest predykatem oznaczającym, że y jest częścią x , a $C(x, y)$ jest predykatem mówiącym, że y jest konstytuentem x . ■

Przykładem jest właściwość *bycia rodzicem*, która jest zależna od właściwości *bycia dzieckiem*. Nie jest natomiast zależna właściwość *bycia człowiekiem*. Jej zależność od *posiadania serca* nie spełnia definicji 3.9, gdyż serce jest częścią ciała człowieka.

Nie jest to definicja kompletna, gdyż zależność między φ a ψ nie może obejmować cech jakościowych (np. kolor), rzeczy, które muszą istnieć (np. wszechświat), oraz sytuacji, gdy φ subsumuje ψ .

Zewnętrznie zależna właściwość jest oznaczana symbolem +D, właściwość, która nie jest zależna zewnętrznie — symbolem –D.

3.1.2 Ograniczenia wynikające z metawłaściwości

Opisane powyżej metawłaściwości pomagają projektantom poprawnie budować zależności taksonomiczne w ramach tworzonych baz wiedzy poprzez narzucenie ograniczeń dotyczących subsumcji:

- 1) $\varphi^{\sim R}$ nie może subsumować $\psi^{\pm R}$.
- 2) $\varphi^{\pm I}$ nie może subsumować $\psi^{\pm I}$.
- 3) $\varphi^{\pm U}$ nie może subsumować $\psi^{\sim U}$.
- 4) $\varphi^{\sim U}$ nie może subsumować $\varphi^{\pm U}$.

- 5) φ^{+D} nie może subsumować ψ^{-D} .
- 6) Właściwości o niekompatybilnym kryterium tożsamości lub całości muszą być rozłączne (nie mogą istnieć encje wykazujące jednocześnie obie właściwości).

Dodatkowo autorzy czynią dwa założenia:

- 1) *Konkretyzacja sortali* (ang. *sortal individuation*): każdy element dziedziny musi wykazywać jakąś właściwość przenoszącą kryterium tożsamości (+I).
- 2) *Rozszerzalność sortali* (ang. *sortal expandability*): jeśli dwie encje wykazują różne właściwości i są tożsame, to muszą wykazywać trzecią właściwość, która przenosi kryterium tożsamości (pozwalające stwierdzić, że są tożsame).

Przestrzeganie powyższych ograniczeń ma wiele zalet. Konstruowanie baz wiedzy zgodnie z nimi sprawi, że będą one bardziej czytelne i mniej w nich będzie przypadków zakłócających ich taksonomiczny charakter (gdy wiele właściwości subsumuje jedną). Przede wszystkim zaś sprawi, że bazy te łatwo będzie rozbudowywać w przyszłości. Jak długo nowo dołączane właściwości będą zachowywały dotychczas przypisane metawłaściwości, tak długo nie trzeba będzie zmieniać dotychczasowych relacji subsumcji. Łatwiejsze też będzie łączenie tych baz z innymi. Szczególną rolę odgrywają właściwości immanentne. Wydzielona część ontologii zawierająca tylko właściwości immanentne nazywa się *taksonomią szkieletową* (ang. *backbone taxonomy*). Właściwości, które ją tworzą (nazwane *właściwościami szkieletowymi* — ang. *backbone properties*), dzielą się na:

- *Kategorie* (ang. *categories*): właściwości nie przenoszące kryterium tożsamości. Ponieważ zgodnie z zasadą 2) nie mogą być subsumowane przez sortale, stanowią najwyższy poziom taksonomii i są prymitywne w takim sensie, że nie narzucają żadnych warunków przynależności. Kategorie tworzą archetypy, takie jak *Rzecz*, *Lokalizacja* itp.
- *Typy* (ang. *types*): właściwości dostarczające kryterium tożsamości. Są to najważniejsze składniki taksonomii, gdyż są jedynymi właściwościami, które dostarczają kryterium tożsamości. Zgodnie z założeniem konkretyzacji sortali każdy składnik dziedziny musi konkretyzować jeden z typów. Ze względu na ograniczenie nr 6) mogą one być subsumowane przez kategorie albo typy, których kryterium tożsamości jest kompatybilne, albo przez właściwości nie będące sortalami. Przykładowymi typami są: *Człowiek*, *Kot*, *Woda*.
- *Quasi-typy* (ang. *quasi-types*): właściwości przenoszące, ale nie dostarczające kryterium tożsamości. Muszą zawsze być subsumowane przez co najmniej jeden typ. Zwykle właściwości tego rodzaju wprowadzają nowe *kryterium przynależności* (ang. *membership condition*). Kryterium tożsamości mówi, na podstawie czego da się odróżnić poszczególne encje wykazujące właściwość będącą sortalem. Kryterium przynależności mówi, co je do siebie upodabnia. Przykładem quasi-typu może być *Student*, *Drapieżnik* itp.

Analiza taksonomii szkieletowej przez osobę niezaznajomioną z daną bazą wiedzy pozwala jej na szybkie rozeznanie głównych podziałów grup obiektów w ramach opisywanej dziedziny. Dzięki temu osłabieniu ulega problem konstrukcyjny.

3.1.3 Dodatkowe metawłaściwości

Oprócz wymienionych w poprzednich punktach metawłaściwości podstawowych autorzy metody OntoClean definiują metawłaściwości pomocnicze.

CO. *Właściwości policzalne* (ang. *countable properties*). Są to właściwości, które przenoszą zarówno kryterium tożsamości (+I), jak i kryterium całości (+U). Pokrywają się one z odniesieniem przedmiotowym rzeczowników policzalnych — każda rzecz, która jest policzalna, musi być odróżnialna i musi być całością. Właściwości +CO nie mogą subsumować właściwości –CO.

ME. *Właściwości mereologicznie ekstensjonalne* (ang. *mereologically extensional*). Obejmują *kolekcje*. Dostarczają one ekstensjonalne kryterium tożsamości (+O), które działa w ten sposób, że dana encja jest wyznaczana przez sumę swoich części, które *konstytuują* całość (są *konstytutentami* całości). Przykładem właściwości mereologicznie ekstensjonalnej jest *bycie kolekcją obrazów* lub *bycie grupą ludzi*. Ten rodzaj właściwości można oznaczyć symbolem +ME. Właściwości +ME nie mogą subsumować właściwości –ME.

UT. Właściwości przenoszące całość topologiczną (patrz punkt 3.1.1.3). Przenoszą kryterium całości (+U). Właściwości +UT nie mogą subsumować właściwości –UT.

UM. Właściwości przenoszące całość morfologiczną (patrz punkt 3.1.1.3). Przenoszą kryterium całości (+U). Właściwości +UM nie mogą subsumować właściwości –UM.

UF. Właściwości przenoszące całość funkcjonalną (patrz punkt 3.1.1.3). Przenoszą kryterium całości (+U). Właściwości +UF nie mogą subsumować właściwości –UF.

3.1.4 Przykłady

Przykład pierwszy ma za zadanie unaocznic, w jaki sposób przyjęte kryterium tożsamości wpływa na decyzję projektową dotyczącą kształtu taksonomii opisującej subsumcję właściwości.

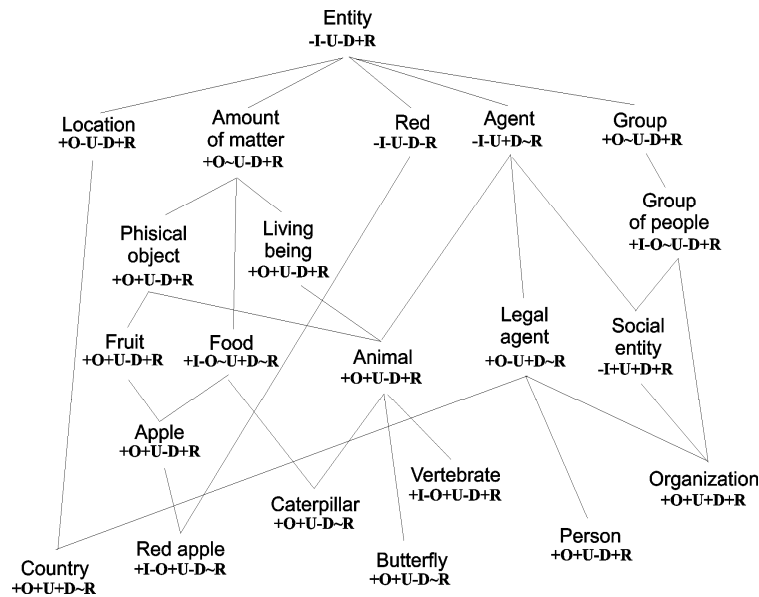
Przykład 3.1 (określenie identyczności)

Zaproponowano wprowadzenie do bazy wiedzy właściwości o nazwie *odstęp czasu*. Encjami wykazującymi tę właściwość byłyby: *godzina*, *dwie godziny*, *kwadrans*. W tej bazie wiedzy istniała już właściwość *odcinek czasu*, jednakże ta ostatnia opisywała konkretny odcinek czasu, np. *od 1:00 do 2:00 w dniu 1.01.2010 roku*. Pojawiło się pytanie: czy *odcinek czasu* powinien być subsumowany przez *odstęp czasu*. Wydawało się to sensownym rozwiązaniem, gdyż, zgodnie z intuicją, każdy odcinek czasu jest też odstępem czasowym. Aby poprawnie odpowiedzieć na to pytanie, należy rozważyć kryteria tożsamości dla obu właściwości. W przypadku *odstępu czasu* dwie encje wykazujące tę właściwość są tożsame, jeśli identyczny jest czas ich trwania. Dwa odstępy czasu trwające równo jedną godzinę są tym samym odstępem czasu. W przypadku *odcinka czasu* kryterium tożsamości jest inne, gdyż musi też być wzięty pod uwagę moment rozpoczęcia odliczania. Odcinek *od 1:00 do 2:00 w dniu 1.01.2010 roku* i odcinek *od 3:00 do 4:00 w dniu 1.01.2010 roku* byłyby dwiema różnymi encjami z punktu widzenia właściwości *odcinek czasu*, a jedną i tą samą encją z punktu widzenia właściwości *odstęp czasu*. Widać więc, że są to dwa różne zbiory encji, a odstęp czasu może być jedynie *komponentem* odcinka czasu. ■

Przykład drugi opisuje proces porządkowania ontologii z wykorzystaniem zaproponowanych zasad.

Przykład 3.2 (porządkowanie ontologii)

Autorzy metody OntoClean prezentują taksonomię, która jest sporządzona na podstawie dostępnych w Internecie ontologii, takich jak WordNet, Pangloss i Mikrokosmos. Jest ona przedstawiona na rys. 3.1.



Rys. 3.1 Przykładowa taksonomia przed uporządkowaniem (za [GW2000d]).

Na rysunku pod nazwami właściwości zamieszczone są ich metawłaściwości. Oto kilka przykładowych decyzji określających te metawłaściwości dla poszczególnych terminów z taksonomii:

Entity

Jest to najbardziej abstrakcyjna właściwość. Nie dostarcza więc ani kryterium tożsamości ($-I$), ani kryterium całości ($-U$), nie jest od niczego zależna ($-D$). Musi jednak być właściwością każdej encji, więc jest immanentna ($+R$).

Location

Właściwość ta określa obszar w przestrzeni. Może być na przykład wyrażona obszarem na mapie. Kryterium tożsamości (ekstensjonalne) jest proste: dwa obszary składające się z tych samych podobszarów są tym samym obszarem ($+O$). Kryterium całości nie jest dostarczane ($-U$). Właściwość nie jest zależna ($-D$). Właściwość jest immanentna, gdyż żadna encja, która jest obszarem, nie może przestać nim być ($+R$).

Amount of matter

Właściwość ta dotyczy pewnej ilości materii określonej masą lub objętością. Podobnie jak w przypadku poprzedniej właściwości, mamy ekstensjonalne kryterium tożsamości ($+O$), brak kryterium całości ($\sim U$), brak zależności od innych właściwości ($-D$) oraz immanentność ($+R$).

Red

Właściwość oznacza *bycie czerwoną rzeczą*, a nie *bycie kolorem czerwonym*. Rzeczy czerwone nie posiadają jednolitego kryterium tożsamości ($-I$) ani jednolitego kryterium

całości (-U). Właściwość ta nie jest też immanentna dla encji, gdyż rzecz czerwona może zmienić kolor (-R).

Agent

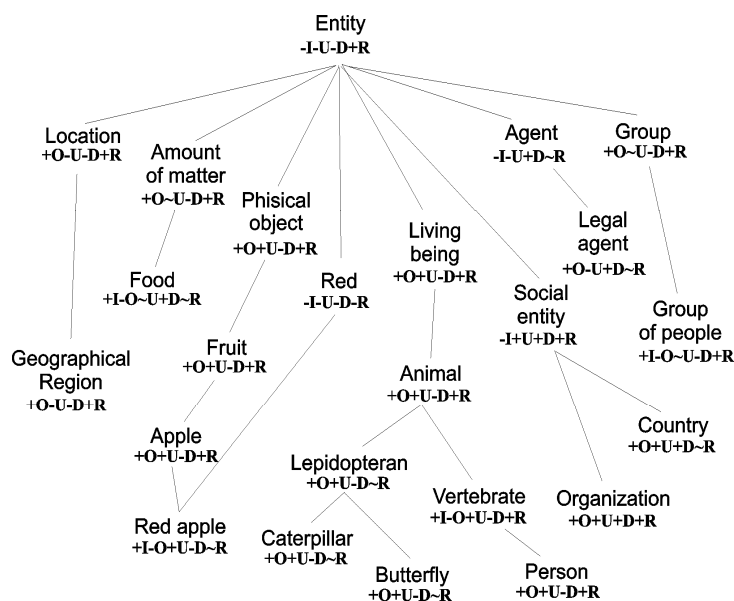
Agentem jest encja, która odgrywa czynną rolę w jakimś zdarzeniu. Prawie każda rzecz może być agentem. Nie istnieją więc jednolite kryteria tożsamości (-I) i całości (-U). Czynność wymaga istnienia obiektu, na którym jest wykonywana, więc właściwość jest zależna (+D). Czynność trwa jedynie ograniczony odcinek czasu, więc właściwość jest antyimmanentna (~R).

Group

Grupa jest nieposiadającą struktury wewnętrznej kolekcją całości. Wystąpienia tej właściwości posiadają ekstensjonalne kryterium tożsamości (+O), co oznacza, że grupa jest definiowana przez elementy ją tworzące. Ponieważ nie ma możliwości wydzielić grupy z innej grupy, żadna grupa nie jest całością (~U). W końcu wygodnie jest założyć, że każde wystąpienie grupy musi być grupą, więc właściwość jest immanentna (+R).

Food

Bycie żywnością nie jest konieczne. Ale też każda rzecz może być żywnością. W sensie językowym *bycie żywnością* jest rolą w zdarzeniu, jakim jest jedzenie. W związku z tym właściwość ta jest antyimmanentna (~R).



Rys. 3.2 Przykładowa taksonomia po uporządkowaniu (za [GW2000d]).

Efektom przydzielenia poszczególnym terminom metawłaściwości oraz zastosowania reguł podanych w punkcie 3.1.2 jest nowy układ taksonomii przedstawiony na rysunku 3.2. Układ ten charakteryzuje się większą przejrzystości i prostotą. Mniej jest na przykład sytuacji gdzie dana właściwość jest subsumowana przez wiele właściwości (wielodziedziczenie). Poza tym tak sformułowany porządek taksonomii będzie bardziej odporny na przyszłe zmiany, które nie będą prowadziły do radykalnych zmian w ramach relacji subsumcji. ■

3.1.5 Metoda OntoClean a projektowanie relacyjnej bazy danych

OntoClean jest czymś więcej niż tylko metodą porządkowania ontologii. Zanim autorzy opublikowali wyniki swoich prac, projektant podejmował swoje decyzje na podstawie intuicji. Autorzy metody wniknęli w głąb procesu konceptualizacji i usystematyzowali pewne zjawiska, które w czasie tego procesu zachodzą. Wyodrębniając cechy właściwości, mówią, jak wpływają one na wzajemne zależności między nimi i jakie w związku z tym będą konsekwencje podjęcia takiej lub innej decyzji projektowej. Zastosowanie tej metody sprawiło, że intuicja ustąpiła miejsca świadomemu wyborowi. Czynność nadawania właściwościom metawłaściwości będzie w dalszej części rozprawy nazywana *metasystematyzacją konceptualną* lub *metasystematyzacją*; powinna ona stać się integralną częścią każdego procesu konceptualizacji.

Metoda OntoClean da się na przykład zastosować do procesu projektowania relacyjnych baz danych. Łatwo dostrzegalną analogią jest podobieństwo kryterium tożsamości do klucza głównego relacji. Formuły (3.2) i (3.3) równie dobrze mogą dotyczyć krotek relacji. Dla dalszych rozważań warto je przytoczyć w uproszczonej, nie zawierającej predykatu egzystencjalnego i czasu formie:

$$\varphi(x) \wedge \varphi(y) \wedge x = y \rightarrow \chi(x, y) \quad (3.4)$$

$$\varphi(x) \wedge \varphi(y) \wedge \chi(x, y) \rightarrow x = y \quad (3.5)$$

Zakładając, że mowa jest o relacji r posiadającej schemat R i klucz główny K (rozumiany jako podzbiór atrybutów R), można zapisać, że:

$$r(t_1) \wedge r(t_2) \wedge t_1 = t_2 \rightarrow t_1[K] = t_2[K] \quad (3.6)$$

$$r(t_1) \wedge r(t_2) \wedge t_1[K] = t_2[K] \rightarrow t_1 = t_2 \quad (3.7)$$

gdzie $t[K]$ oznacza zbiór składowych krotki t dla wszystkich atrybutów z K .

Mówiąc mniej formalnie, jeśli dwie krotki relacji r są sobie równe, to równe są ich składowe związane z kluczem K (warunek konieczny). I odwrotnie, jeśli dwie krotki relacji r mają równe składowe związane z kluczem K , to są sobie równe (warunek wystarczający). Można przyjąć, że w tym wypadku kryterium tożsamości krotek $\chi(t_1, t_2)$ jest równość składowych $t_1[K] = t_2[K]$.

Gdyby dodać do formuł (3.6) i (3.7) warunek nieredukowalności K , to w pełni odpowiadałyby one definicji klucza kandydującego obowiązującej w modelu relacyjnym. Ustanowienie klucza głównego to podjęcie decyzji wyboru: które z dostępnych kryteriów tożsamości przyjmuje się jako obowiązujące. Na przykład gdyby relację *Ludzie* uznać za odpowiednik właściwości *bycia człowiekiem*, to można przyjąć, że kluczem głównym może być atrybut `numerPesel`. Relacja ta dostarczałaby to kryterium tożsamości (+O), więc byłaby też immanentna (+R). Immanentność nie oznacza w przypadku relacyjnej bazy danych, że krotki nie można usunąć. Należy ją rozważyć w konfrontacji z jakąś właściwością nieimmanentną. Właściwością nieimmanentną jest na przykład *bycie studentem*. Odpowiednikiem tej właściwości może być relacja *Studentci*¹¹. Aby w bazie relacyjnej uzależnić dwie relacje od siebie, łączy się je kluczem obcym. Zgodnie z intuicją relacja *Studentci* powinna być powiązana z relacją *Ludzie* kluczem obcym opartym na atrybucie `numerPesel`. Jak widać relacja ta nie dostarcza nowego kryterium tożsamości, lecz jedynie je przenosi (+I).

¹¹ Stwierdzenie, że relacja *Studentci* jest odpowiednikiem właściwości *bycia studentem* jest oczywiście daleko idącym uproszczeniem. Sposób opisywania świata za pomocą jedynie predykatów unarnych sprawia, że każdy atrybut tej relacji jest również właściwością. Na przykład zaproponowany dalej atrybut `numerPesel` jest odpowiednikiem właściwości *posiadania numeru PESEL*.

Zaczerpnięte z metody OntoClean metawłaściwości pozwalają uzasadnić ograniczenia tworzone wewnątrz opisywanej bazy danych. Takim ograniczeniem jest na przykład kaskadowe usuwanie krotek. Z faktu, że relacja *Ludzie* jest immanentna, a relacja *Studenci* taka nie jest, wynika, że usunięcie krotki z tej pierwszej relacji musi spowodować także usunięcie wszystkich odpowiadających krotek z tej drugiej. Natomiast ustanowienie podobnej zasady działającej w przeciwną stronę nie byłoby właściwe. Odpowiadałoby ono w bazie wiedzy przyjęciu rozwiązania mówiącego, że właściwość *bycia studentem* subsumuje właściwość *bycia człowiekiem*, a tego zabrania ograniczenie nr 1) z rozdziału 3.1.2.

Jak widać z powyższych rozważań, metoda OntoClean dostarcza uniwersalnych zasad związanych z procesem konceptualizacji. Można jej językiem wyrazić różne problemy charakterystyczne dla różnych metod modelowania rzeczywistości. Dotyczy to zarówno modelu ontologicznego, jak relacyjnego i obiektowego. Jej wartość przejawia się wszędzie tam, gdzie projektant podczas modelowania posługuje się relacją subsumcji, której kształt, wynikający dotąd z intuicji, staje się wynikiem znormalizowanego procesu.

3.2 Zastosowanie ontologii fundacjonistycznych

Koncepcja podziału ontologii na warstwy zawierające opisy o różnych poziomach abstrakcji powstała na przełomie XX/XXI wieku. Jej propagatorami byli m.in. J. Sowa [S1995] i N. Guarino [G1997]. Opierała się na spostrzeżeniu, że każdy z tworzonych opisów, oprócz warstwy opisowej ściśle związanej z daną dziedziną, musi też zawierać pewną warstwę ogólną, która jest wspólna dla wszystkich dziedzin. Istnieje więc możliwość wydzielenia tej warstwy i udostępnienia jej wszystkim chętnym w postaci przypominającej bibliotekę standardowych funkcji udostępnianą programistom. Intuicja podpowiadała, że dzięki takiemu podziałowi uda się usprawnić tworzenie ontologii dziedzinowych, gdyż inżynierowie zostaną zwolnieni z obowiązku definiowania terminów podstawowych. Oprócz tego wierzone, że użycie standardów wymusi na twórcach ontologii stosowanie określonego punktu widzenia. Wszystkie ontologie dziedzinowe oparte na tej samej warstwie ogólnej staną się do siebie podobne. To ułatwi powtórne użycie, gdyż łączenie ontologii mogłoby odbywać się poprzez tłumaczenie terminów dziedzinowych zgodnie z językiem definiowanym na wyższym poziomie. Tak więc problem integracyjny mógłby zostać w znacznym stopniu złagodzony. Kolejnymi zaletami takiego podejścia miało być zmniejszenie ontologii dziedzinowych i ich uproszczenie. Pozbawione warstwy wiedzy ogólnej koncentrowałyby się wyłącznie na problemach dziedzinowych. Mniejsze i prostsze ontologie byłyby oczywiście łatwiejsze do zrozumienia i zaprojektowania, co dawało możliwość złagodzenia problemów konstrukcyjnego i organizacyjnego.

Koncepcji podziału wiedzy na warstwy towarzyszy wizja stworzenia w Internecie wielkiej, hierarchicznie uporządkowanej sieci powiązanych ze sobą ontologii. Na najwyższym poziomie hierarchii znajdowałyby się ontologie wysokiego poziomu (ang. *top-level ontologies*). Czasami ontologie te są nazywane *ontologiami fundacjonistycznymi*¹² (ang. *foundational ontologies*). W kolejnej warstwie usytuowane byłyby ontologie średniego poziomu (ang. *mid-level ontologies*). Najniższy poziom tej hierarchii to ontologie dziedzinowe (ang. *domain ontologies*). Każda z tych warstw miałaby przypisaną inną rolę. Ontologie wysokiego i średniego poziomu byłyby dziedzinowo niezależne i definiowałyby pojęcia kojarzone z powszechną wiedzą (ang. *common sense*¹³), czyli takie, które są potrzebne człowiekowi do rozumienia świata. Te pierwsze obejmowałyby koncepty podstawowe

¹² Nazwy *ontologia fundacjonistyczna* w polskim opracowaniu użyto po raz pierwszy w [CP2007].

¹³ Najczęściej angielski *common sense* jest tłumaczony na *zdrowy rozsądek*. Wg autora tej pracy tłumaczenie *wiedza powszechna* w kontekście ontologii fundacjonistycznych jest właściwsze.

i uniwersalne, na przykład definiowałyby takie pojęcia, jak *klasa*, *obiekt*, *proces*, *lokalizacja*, itp. Te drugie opisywałyby już rzeczy bardziej konkretne. Na przykład rozszerzałyby pojęcie *lokalizacji*, definiując *obszar geograficzny*.

Idea znalazła wielu zwolenników i aktywnych uczestników. Od końca lat 90. ubiegłego wieku powstała znaczna liczba ontologii fundacjonistycznych, np. DOLCE (<http://www.loa-cnr.it/DOLCE.html>), SUMO (<http://www.ontologyportal.org/>), GFO (<http://www.onto-med.de/ontologies/gfo/index.jsp>). Oto ich krótka charakterystyka (na podstawie [MCR2006] i [SPOP2004]).

Ontologia DOLCE powstała w trakcie trwania projektu WonderWeb finansowanego przez Komisję Europejską, czyli w latach 2002-2004. Jest niedużą ontologią, składa się z ok. 100 konceptów i ról, i tej samej liczby aksjomatów. Najbardziej ogólnym terminem jest *Particular*. Dzieli się on na koncepty *Abstract* i *Spatio-temporal-particular*. Od tego ostatniego dziedziczą *Endurant*, *Perdurant* i *Quality*. Ze względu na nazwę konceptu najbardziej ogólnego, autorzy DOLCE nazywają ją „ontologią partykulariów”, czyli ontologią obiektów, a nie terminów i właściwości ogólnych. Ontologia DOLCE jest wciąż rozwijana i ze względu na swoją kompaktowość cieszy się dużą popularnością.

Ontologia SUMO jest starsza niż DOLCE. Powstała w 2000 roku. Jest też dużo większa: zawiera ok. 1000 terminów i ok. 4000 formuł. Jest to ontologia modułarna, składa się z 11 modułów powiązanych wzajemnie zależnościami. Konceptem najbardziej ogólnym jest w niej *Entity*. Dzieli się on na *Physical* i *Abstract*. Pierwszy z nich dzieli się na *Object* i *Process*, a drugi na *SetClass*, *Proposition*, *Quantity* i *Attribute*. Ontologia SUMO posiada własną ontologię średniego poziomu, która nazywa się MILO.

Prace nad ontologią GFO rozpoczęły się w 1999 roku i jest ona do dziś doskonała. W jej słowniku znajduje się ok. 250 terminów. Na szczycie znajduje się koncept *Entity*, od którego dziedziczą *Set* i *Item*. Koncept *Item* dzieli się na *Individual* i *Category*. Z kolei *Individual* dzieli się na *Abstract*, *Concrete* i *Space-time*.

Osobno warto wymienić ontologię CYC (<http://www.cyc.com/>), ponieważ powstała najwcześniej (w 1984 roku, na długo przed powstaniem innych) i jest największa. Zawiera 300 000 konceptów i 3 000 000 formuł. Jest to także ontologia kontekstowa, oparta na formalizmie PLC, który będzie opisany w punkcie 4.1.3.

3.3 Podsumowanie

Powyższy rozdział przedstawił dwie metody normalizacji baz wiedzy: metodę OntoClean oraz metodę polegającą na zastosowaniu ontologii fundacjonistycznych.

Ocena przedstawionych metod nie jest jednoznaczna. Jako sposób na rozwiązanie wymienionych we wstępie do niniejszej rozprawy problemów referencyjnych, miały one przede wszystkim wspomóc etap projektowania baz wiedzy. Zarówno metoda OntoClean, jak i ontologie fundacjonistyczne mają na celu narzucenie twórcom baz wiedzy pewnych ograniczeń, dzięki którym tworzone taksonomie miałyby właściwą strukturę. Właściwa struktura miała pomóc w rozwiązaniu problemu konstrukcyjnego, pozwalając na łatwe tworzenie ograniczonych widoków w celach prezentacyjnych. Podobieństwo ujęcia tych samych zjawisk mogło specjalistom, wraz z nabywaniem doświadczenia, ułatwić ogarnięcie struktur modeli i zachowanie nad nimi kontroli. W sposób radykalny miał być rozwiązany problem integracyjny. Właściwa struktura tworzonych według jednego wzorca ontologii miała umożliwiać łatwe ponowne wykorzystanie oraz integrację ([GW2000c]).

W praktyce osiągnięcie tych celów jest wysoce problematyczne. Główną wadą stosowania normalizacji jest definitywność ustalonego punktu widzenia. Metoda OntoClean narzuca konieczność nadania konceptom metawłaściwości, które raz nadane, narzucają porządek relacji subsumcji nakładanej na te koncepty. Usztywnia to opis świata, a jednocześnie nie gwarantuje w najmniejszym stopniu, że będą dzięki temu powstawały ontologie opisujące świat w sposób bardziej homogeniczny. Można to zobrazować na przykładzie 3.2, w którym autorzy dokonują metasytematyzacji danej ontologii. W jej wyniku koncept *Food* uzyskuje metawłaściwości, które uniemożliwiają dziedziczenie od niego takich konceptów, jak *Fruit*. Wychodzi więc na to, że ontologii tej nie da się w przyszłości rozbudować w stronę opisu takich tematów, jak podział gatunków na *roślinożerne* i *drapieżniki*, jak rozwinięcie pojęć *zalecana dieta* lub *zdrowa żywność* itp. Z pewnością zdarzy się, że inna ontologia, wykonana również zgodnie z zaleceniami metody OntoClean, ujmie pojęcie żywności właśnie od tej strony i wtedy trudność powtórnego wykorzystania nie będzie ani trochę mniejsza niż w przypadku ontologii wykonanych bez stosowania normalizacji.

Stosowanie ontologii fundacjonistycznych w niewielkim tylko stopniu poprawia sytuację. Ontologie te tworzone są na podstawie opracowań filozoficznych, a te jak wiadomo różnią się między sobą. Na przykład znany jest tzw. spór o uniwersalia. Stanowisko określane jako *realizm ontologiczny* zakłada, że rzeczy (w tym pojęcia ogólne) istnieją niezależnie od ludzkiego sposobu postrzegania, myślenia i porozumiewania się (języka) [H2004]. *Konceptualizm* z kolei zakłada, że pojęcia istnieją tylko w ludzkich umysłach. W końcu *nominalizm* twierdzi, że pojęcia są jedynie elementami języka. W efekcie różne ontologie fundacjonistyczne dokonują różnej konceptualizacji. Nie jest też łatwa rola inżyniera, który, korzystając z nich, próbuje stworzyć ontologię dziedzinową. Na przykład: czy koncept *algorytm* powinien być zaliczony do pojęcia *odkryty* (czyli niezmienny w czasie — w ontologii DOLCE odpowiada temu koncept *Endurant*), czy pojęcia *stworzony* (czyli zmienny w czasie — w ontologii DOLCE odpowiada temu koncept *Perdurant*)? Bez głębszej znajomości filozofii inżynier może podejmować niekonsekwentne decyzje, co utrudni, a może nawet uniemożliwi, ponowne użycie takiej ontologii.

Pomimo wymienionych wad rozwinięcie metod normalizacji wniosło wielki wkład do poznania samego procesu konceptualizacji. Ontologie fundacjonistyczne same w sobie stanowią konkretyzację teorii filozoficznych i poznanie ich pomaga inżynierom lepiej opisywać rzeczywistość. Szczególną rolę odgrywają jednak prace Guarino i Welty'ego związane z opracowaniem metody OntoClean. Dzięki nim stworzone zostały teoretyczne podstawy opisu procesu konceptualizacji, uświadomiona została natura procesów, które były dotąd domeną intuicji. Wbrew pozorom prace te rzuciły też światło na kontekstową naturę każdego opisu, wiążąc go z pojęciami zaczerpniętymi z logiki modalnej. Otóż zwykła ontologia jest tworem bezkontekstowym, gdy spojrzysz na nią jak na twór statyczny. Autorzy metody OntoClean, sięgając do pojęcia światów możliwych, potraktowali ontologię jak twór dynamiczny. Przypisanie konceptowi immanentności oznacza, że osobnik, który jest wystąpieniem tego konceptu, zawsze już nim pozostanie. Zawsze, czyli w każdym świecie możliwym, oznacza w tym przypadku: w każdej wersji projektowanej ontologii. Należy więc patrzeć na ontologię jako na twór zmieniający się w czasie. Chodzi tu oczywiście o zmiany niemonotoniczne. Gdyby bowiem ontologia miała być monotoniczna, każde zdanie wprowadzone do niej (na przykład asercja *Student(jan_kowalski)*) musiałoby już na zawsze w niej pozostać. Nie miałby wtedy sensu podział na koncepty immanentne i nieimmanentne, gdyż osobnik *jan_kowalski* na zawsze musiałby już zostać studentem, czyli każda właściwość byłaby immanentna. Zmiany niemonotoniczne, które przewiduje projektant, zastanawiając się nad immanentnością, to właśnie dostępne dla niego światy możliwe. Semantykę operatora modalnego \square opisuje wyrażenie (4.2) przedstawione

w podrozdziale 4.1. Dane zdanie jest koniecznie prawdziwe wtedy i tylko wtedy, gdy jest uznawane za prawdziwe we wszystkich dostępnych światach możliwych. Logika modalna nie precyzuje pojęcia światów możliwych, choć próby wyjaśnienia, czym mogą one być z metafizycznego punktu widzenia, są podejmowane [C1979]. Nie ulega jednakże wątpliwości, że można je utożsamiać z różnymi sytuacjami. Różne sytuacje wymuszają różne punkty widzenia, tworzą różne konteksty. Podczas projektowania twórca ontologii musi wyobrazić sobie różne sytuacje, w jakich może znaleźć się użytkownik. Wszystkie dostępne z jego punktu widzenia światy możliwe muszą „mieścić w sobie” wszystkie światy możliwe dostępne dla użytkownika. Gdyby tak nie było, to znaczy: gdyby twórca założył, że jakiś użytkownik rozwinie jego ontologię w takim kierunku, że dana encja utraci właściwość, która została przez niego uznana za immanentną, to musiałby również uznać, że ta właściwość wcale immanentna nie jest. Tak więc, projektując ontologię, twórca musi rozwinąć wyobraźnię kontekstową, musi dokonać konceptualizacji, nie abstrahując od tych różnych wyobrażonych kontekstów.

4 Modularyzacja ontologii

Drugim obok normalizacji kierunkiem, w którym podąża rozwój metod mających na celu przełamanie ograniczeń związanych z szerokim zastosowaniem ontologii, jest modularyzacja. Ogólnie mówiąc, terminem *modularyzacja* obejmuje się szeroko pojęte działania mające na celu wyodrębnianie fragmentów ontologii oraz łączenie tych fragmentów w sposób jak najbardziej adekwatny do postawionego zadania. Wśród tych działań te, które są w tym rozdziale szczególnym przedmiotem zainteresowania, wyznaczane są przez grupę metod opierających się na założeniu, że ontologia jest tworzona i użytkowana jako twór modułarny, że modularność jest jej cechą immanentną. Wiedza podzielona na moduły jest bardziej czytelna i zrozumiała, a także pozwala ograniczyć zasięg przeprowadzanego wnioskowania jedynie do informacji najbardziej adekwatnej ze względu na rozwiązywany problem. Poza tym modularne, kontekstowe ujęcie wiedzy zgadza się z wynikami badań psychologicznych nad organizacją ludzkiej pamięci oraz sposobem, w jaki umysł z niej korzysta.

Większość prac związanych z modularyzacją wychodzi od pojęcia kontekstu. Jest to zrozumiałe, gdyż ontologie są twórcami logicznymi, a logika ma swoje źródła w języku naturalnym. Język naturalny jest językiem kontekstowym. Formalizmy logiczne starają się odwzorować maksymalnie dużo zjawisk językowych, dlatego już od dawna starano się uwzględnić wpływ, jaki kontekst wypowiedzi ma na treść wypowiedzianych zdań. Mówi o tym podrozdział 4.1. Przedstawiono w nim kilka znaczących propozycji uwzględnienia w logice istnienia kontekstów. Propozycje te stały się teoretycznym fundamentem dla technologii modułarnych opracowanych specjalnie na potrzeby inicjatywy Semantic Web. Opisuje je kolejny podrozdział, 4.2. Wobec opisanych we wstępie do niniejszej pracy problemów referencyjnych, modularyzacja wydawała się panaceum na wszelkie dolegliwości. W podrozdziale 4.3 zamieszczone jest podsumowanie, a w nim ocena przedstawionych metod z punktu widzenia stawianych przed nimi celów.

4.1 Konteksty w logice

Przekonanie o konieczności uwzględnienia kontekstów w logice nie jest nowe. Wynika ono ze spostrzeżenia, że poszczególne słowa, zwroty czy nawet fragmenty wypowiedzi formułowanych w języku naturalnym mają zmienne znaczenie. Mówi się, że znaczenie danej wypowiedzi jest zależne od kontekstu. Jednakże samo słowo *kontekst* nie ma ścisłej definicji. Najbardziej podstawowa jest jego definicja lingwistyczna. „Słownik wyrazów obcych” Władysława Kopalińskiego [K2000] definiuje je jako „tekst, w którym użyto danego (zwłaszcza cytowanego gdzie indziej) wyrazu albo zwrotu”.

„Słownik języka polskiego” PWN [S1978] podaje cztery znaczenia, z których jedno wychodzi poza terminy lingwistyczne: „zespół czynników współlistniejących, powiązanych z czymś”. To „coś” może być sytuacją, otoczeniem, dziedziną, układem, tłem lub środowiskiem i zawierać encje, osobniki lub cokolwiek innego, co jest przedmiotem zainteresowania.

Początkowo logicy byli zainteresowani wąskim rozumieniem słowa kontekst. W końcu XIX w. jeden z twórców rachunku predykatów, Amerykanin Charles Sanders Peirce, opracował graficzną reprezentację formuł logicznych uwzględniającą istnienie kontekstów lingwistycznych (za [S2000]). Były one oznaczane zawierającymi się w sobie owalami, które obejmowały grafy fragmentów zdań. Głównym celem tej notacji było odwzorowanie takich zjawisk językowych, jak zaimki lub polinomie.

Rozwój logiki i jej zastosowań spowodował, że konieczne stało się sformalizowanie takich zjawisk, jak występowanie różnych punktów widzenia, czy istnienie różnych przekonań (postaw *propozycyjalnych*). Pierwsze z tych zjawisk występuje w procesie dowodzenia twierdzeń, stąd jego formalizacja stała się czynnikiem warunkującym rozwój systemów dowodzenia. Konieczność formalizacji drugiego z tych zjawisk ujawniła się podczas prac nad systemami agentowymi. Ogólnie mówiąc, logicy dostrzegli, że istnieje potrzeba uwzględnienia różnego sposobu wartościowania danego zdania nie tylko w ramach różnych interpretacji, lecz w ramach jednej interpretacji. Formalizmem, który jako pierwszy uwzględnił tę potrzebę, jest *logika modalna*. Logikę tę opracował na początku XX wieku Clarend Irving Lewis. Wprowadza ona tzw. modalności *aletyczne*: *konieczne* i *możliwe*. Początkowo jednak nie zdobyła ona popularności, gdyż nie powstał jednolity system interpretowania operatorów modalnych (konieczne, że p : $\Box p$; możliwe, że p : $\Diamond p$). W latach 60. XX wieku Saul Kripke wprowadził pojęcie światów możliwych, które pozwoliło taką interpretację nadać. Zaproponował on aby model był strukturą składającą się z trzech komponentów:

- zbioru $K = \{w_i \mid 0 \leq i \leq n\}$ światów możliwych z jednym światem uprzywilejowanym $w_0 \in K$, reprezentującym świat rzeczywisty,
- relacji dostępu $R \subseteq K \times K$, która mówi, że dla $(w_i, w_j) \in R$ świat w_j jest dostępny ze świata w_i ,
- funkcja wartościująca $v: P \times K \rightarrow \{0, 1\}$, gdzie P jest zbiorem zdań atomowych, która nadaje zdaniu $p \in P$ wartość logiczną 0 (fałsz) lub 1 (prawda) w świecie $w \in K$.

Wprowadzenie pojęcia światów możliwych pozwoliło na zinterpretowanie operatorów modalności. Dla $p \in P$:

$$\Diamond p \text{ wtwg } \exists w \in K ((w_0, w) \in R \wedge v(p, w) = 1) \quad (4.1)$$

$$\Box p \text{ wtwg } \forall w \in K ((w_0, w) \in R \rightarrow v(p, w) = 1) \quad (4.2)$$

Powyższe formuły można przetłumaczyć na język naturalny w następujący sposób: „Możliwe, że zdanie p jest prawdziwe (ma wartość logiczną prawdy w świecie rzeczywistym) wtedy i tylko wtedy, gdy wśród dostępnych światów możliwych istnieje taki świat w , w którym zdanie p jest uznawane za prawdziwe (ma w świecie w przyporządkowaną wartość logiczną prawdy)” oraz „Zdanie p jest koniecznie prawdziwe wtedy i tylko wtedy, gdy jest uznawane za prawdziwe we wszystkich dostępnych światach możliwych”.

W tym samym czasie Jaakko Hintikka zaproponował system zwany dziś *systemem modelowym Hintikki* (ang. *Hintikka's model system*) ([H1969], [H1992]). W systemie tym wprowadzone zostały pojęcia modalności *epistemicznych*, czyli związanych z możliwościami poznawczymi danego podmiotu. Są to między innymi:

- $\mathbf{P}_x p$ — możliwe, że zdanie p jest prawdziwe ze względu na wiedzę podmiotu x ,
- $\mathbf{K}_x p$ — podmiot x wie, że zdanie p jest prawdziwe.

Modelem w systemie Hintikki jest rodzina \mathcal{M} zbiorów formuł, zwanych *zbiorami modelowymi*. Zbiory te reprezentują różne punkty widzenia. Każdy ze zbiorów zawiera zdania, które zgodnie z danym punktem widzenia mają przyporządkowaną wartość prawdy. Semantykę każdego ze zbiorów $M \in \mathcal{M}$ tworzą reguły, będące odpowiednikami reguł znanych z klasycznego rachunku zdań, określające przynależność formuł. Na przykład odpowiednikiem reguły wyłączzonego środka jest reguła mówiąca, że jeśli $p \in M$, to $\neg p \notin M$. Istnieją także reguły definiujące znaczenie wyżej wymienionych modalności. Tak więc na przykład:

Jeżeli $\mathbf{P}_x p \in M$, to $\exists M' \in \mathcal{M} (p \in M')$ (4.3)

Jeżeli $\mathbf{K}_x p \in M$, to $\forall M' \in \mathcal{M} (\mathbf{K}_x p \in M')$ (4.4)

Z porównania formuł (4.1) i (4.2) oraz (4.3) i (4.4) widać, że istnieje związek między zbiorami modelowymi i światami możliwymi. Można ten związek wyrazić w następujący sposób:

$$\forall M \in \mathcal{M} (\exists w \in K (p \in M \text{ wtvg } w \models p))$$

Tłumacząc to zdanie na język naturalny, można powiedzieć, że dla każdego zbioru modelowego M istnieje taki świat możliwy w , dla którego zdanie p należy do zbioru modelowego M wtedy i tylko wtedy, gdy wynika ona ze świata w .

Wprowadzenie powyższych sposobów interpretacji pozwoliło logikom ująć wiele zjawisk dotychczas nieuchwytnych i w rezultacie umożliwiło reprezentowanie różnych punktów widzenia i różnych postaw propozycjonalnych. Znalazły one uznanie w takich dziedzinach, jak logika dowodliwości (ang. *provability logic*) czy w pracach nad systemami agentowymi. Jednak wielu badaczy problematyki sztucznej inteligencji uznało to rozwiązanie za niewystarczające i obarczone wadami. Konstrukcje, jakimi są światy możliwe i zbiory modelowe, pozwalają wyrazić różne modele świata, ale modele te tyczą się światów jako całości. To daje możliwość opisania np. zbioru systemów agentowych, z których każdy dysponuje inną bazą wiedzy o danej dziedzinie. Nie istnieje jednak możliwość ukazania tej dziedziny z różnych punktów widzenia w ramach wiedzy jednego agenta. Nie istnieje także możliwość nadania tej bazie wewnętrznej struktury, która porządkowałaby ją adekwatnie do rozwiązywanych problemów.

W kolejnych punktach tego podrozdziału zostaną przedstawione różne formalizmy logiczne, które wprowadzają kontekst do logiki jako integralny element systemu. Pierwszy punkt, 4.1.1, odgrywa wyjątkową rolę. Zawiera on ogólną refleksję na temat logik kontekstowych, próbuje określić w jaki sposób formalizmy te powinny ujmować sprawę kontekstów, jakimi zasadami powinny się kierować. Opisane tu ujęcie problematyki kontekstów stanowi ważną inspirację, którą kierował się autor niniejszej rozprawy. W punkcie 4.1.2 przedstawiona jest logika zaimków wskazujących, która była jedną z pierwszych logik wprowadzających do systemu takie elementy określające kontekst, jak czas, miejsce, agent. W punkcie 4.1.3 opisany jest system PLD, który stanowi formalną podstawę rozwiązań stosowanych w ontologii CYC. Punkt 4.1.4 przedstawia system LMS/LCS, który wprowadza szereg rozwiązań zaadaptowanych w ramach głównych technik modularyzacji opracowanych dla inicjatywy Semantic Web.

4.1.1 Wprowadzenie pojęcia kontekstu do logiki

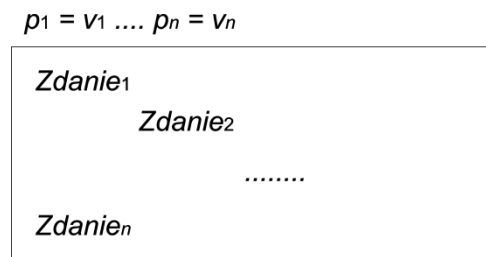
Wprowadzenie kontekstu do logiki łączy się z koniecznością udzielenia odpowiedzi na wiele pytań. Ogólnie cztery najważniejsze z nich to:

- 1) Czym jest kontekst rozumiany jako element języka?
- 2) Co jest umieszczone wewnątrz kontekstu?
- 3) Co znajduje się poza kontekstem?
- 4) W jaki sposób to, co znajduje się poza kontekstem, wpływa na sposób interpretowania wnętrza kontekstu?

Najczęściej proponowaną odpowiedzią na pierwsze pytanie jest przyjęcie, że kontekst jest miejscem interpretowania zdań, czyli nadawania im wartości logicznej. Jednakże odpowiedzi na kolejne pytania nie są już zgodne.

Przyjęcie pewnego uzgodnienia, które w sposób możliwy do zaakceptowania przez większość ludzi ujmie pojęcie kontekstu, jest z punktu widzenia niniejszej rozprawy rzeczą istotną. Jedną z tych pomocniczych mówi bowiem o rozwiązaniu problemu adekwatności kontekstów i problemu projektowania kontekstowego. Zdaniem autora za takie uzgodnienie należy przyjąć propozycję przedstawioną w [BBG2000]. Propozycja ta jest efektem analizy wielu różnych formalizmów w różny sposób rozwiązujących problem kontekstowości, stąd cechuje się trafnością, a jednocześnie ogólnością ujęcia. Można przyjąć, że dane rozwiązanie tym lepiej rozwiązuje oba przytoczone problemy, im lepiej odpowiada postulatowi zawartemu w tym opracowaniu.

W [BBG2000] przedstawiono kontekst jako pudełko (patrz rys. 4.1.).

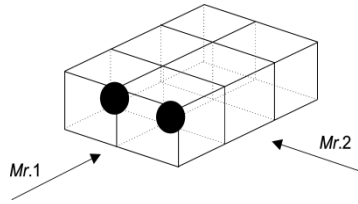


Rys. 4.1 Kontekst jako pudełko (na podst. [BBG2000]).

Wewnątrz kontekstu-pudełka znajdują się zdania. Poza pudełkiem znajdują się parametry p_1, \dots, p_n . Parametrom tym przypisane są wartości v_1, \dots, v_n . Dzięki takiemu opisowi można abstrahować od całej złożoności sytuacji na zewnątrz kontekstu. Wewnątrz kontekstu znajdują się zdania; co do tego nie istnieją kontrowersje. Jednak istnieją różnice w poglądach na to, jakiego rodzaju są to zdania. Mogą to być zdania wspólne dla wszystkich kontekstów (jak zmienne o globalnym zasięgu), a tylko różnie w różnych kontekstach interpretowane. Mogą to być zdania ściśle związane z jednym kontekstem (jak zmienne o lokalnym zasięgu). Język, w którym zdania zostały wypowiedziane, może być językiem wspólnym dla wszystkich kontekstów lub językiem odrębnym dla każdego kontekstu. Jeśli chodzi o listę parametrów, to kontrowersje są znacznie większe. Nie ma oczywiście zgody co do tego, jakie informacje powinny te parametry opisywać. Zawierają one informację, która niejako została „odjęta” od informacji zawartej w zdaniach znajdujących się wewnątrz kontekstu. Przykładowo wartość logiczna zdania: „W pobliżu znajduje się leśniczówka” jest uzależniona od miejsca, gdzie to zdanie zostało wypowiedziane. Podobnie jest ze zdaniem: „Najnowszym odkryciem w dziedzinie transplantologii jest przeszczep twarzy”, tylko że tym razem jego wartość logiczna jest uzależniona od chwili, w której je sformułowano. Te dwa przykłady skłaniają do tego, żeby w parametrach zewnętrznych umieścić współrzędne przestrzenno-temporalne. Jednak parametrów tego rodzaju może być znacznie więcej. Lista parametrów może być skończona lub nieskończona, jawna lub niejawna. Może być taka sama (w sensie nazwy i typu) dla wszystkich kontekstów lub dla każdego inna.

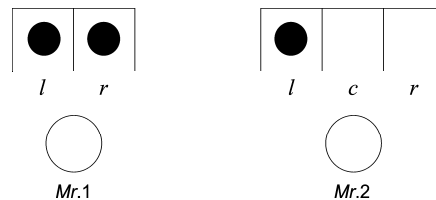
Przykład 4.1 (Magic box)

Prostą ilustracją opisanych wyżej idei może być tzw. magiczne pudełko (ang. *magic box*), przedstawione na rys. 4.2. Dwaj obserwatorzy, nazwijmy ich Mr.1 i Mr.2, obserwują magiczne pudełko z dwóch różnych punktów widzenia. Pudełko jest „magiczne”, ponieważ obserwatorzy nie są świadomi jego głębokości. Rys. 4.3. przedstawia, co widzą obserwatorzy Mr.1 i Mr.2, jeśli ich położenie jest takie, jak pokazano na rys. 4.2.



Rys. 4.2 Magiczne pudełko (za [BBG2000]).

Widoki postrzegane przez obserwatorów Mr.1 i Mr.2 mogą być traktowane jako dwa różne konteksty. Każdy z obserwatorów ma własną, lokalną reprezentację pudełka, która zależy od perspektywy, z której patrzy. Przykładowo Mr.1 widzi dwie kulki, a Mr.2 tylko jedną. Z kolei wiedza, którą dysponuje Mr.2, zawiera pojęcie sekcji centralnej, podczas gdy wiedza Mr.1 takiego pojęcia nie zawiera. Jednakże oba obrazy są od siebie uzależnione. Jeśli dla przykładu jeden z obserwatorów nie widzi żadnej kulki, drugi również nie może dostrzegać żadnej.

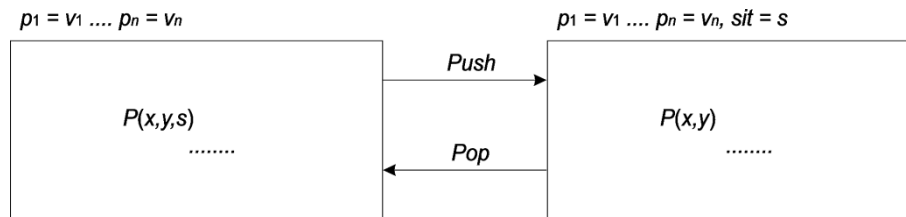


Rys. 4.3 Widok magicznego pudełka postrzegany przez obserwatorów Mr.1 i Mr.2 (za [BBG2000]).

W języku obserwatora Mr.1 istnieją zdania l i r , które reprezentują fakt, że z punktu widzenia tego kontekstu „kulka znajduje się w lewej sekcji” i fakt, że „kulka znajduje się w prawej sekcji”. Analogicznie, w kontekście Mr.2 istnieją zdania l , c i r , które reprezentują fakty: „kulka znajduje się w lewej sekcji”, „kulka znajduje się w centralnej sekcji” i „kulka znajduje się w prawej sekcji”. Obserwacje dokonane w Mr.1 i Mr.2 mogą być sformalizowane przez dwa modele logiki zdań, odpowiednio $\{l, r\}$ i $\{l, c, r\}$. Wartościowanie tych zdań w każdym kontekście jest zrealizowane lokalnie, jednak nie jest niezależne od wartościowania w drugim kontekście. Łatwo zauważyć, że jeśli w kontekście obserwatora Mr.1 chociaż jedno ze zdań jest prawdziwe, to w kontekście obserwatora Mr.2 również co najmniej jedno ze zdań musi być prawdziwe. Występuje także zależność odwrotna. ■

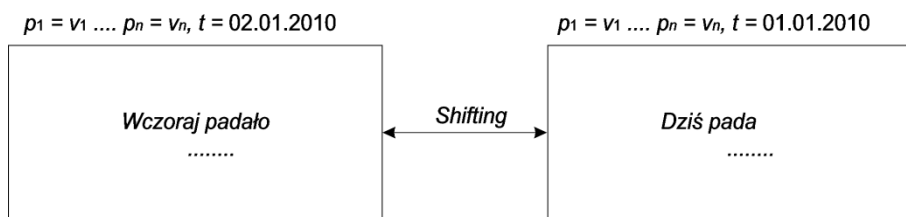
Podstawowym celem stosowania kontekstów jest umożliwienie ograniczenia wiedzy branej pod uwagę podczas rozwiązywania konkretnego problemu. Chodzi więc głównie o *wnioskowanie lokalne*. Jednakże nie ulega kwestii, że to lokalne wnioskowanie musi być uzależnione od informacji zewnętrznej reprezentowanej przez parametry p_1, \dots, p_n . Autorzy [BBG2000] wyodrębniają dwie metody uwzględniania informacji zewnętrznej: *wstawianie i usuwanie* (ang. *push and pop*) oraz *przesuwanie* (ang. *shifting*). Pierwsza z nich zakłada, że istnieje jakiś stan „bezkontekstowy”, w którym lista parametrów jest pusta. Jest to stan idealny, który w praktyce nigdy nie istnieje. Proces *kontekstualizacji* polega na stopniowym usuwaniu ze zbioru zdań kolejnych informacji i umieszczaniu ich w zbiorze parametrów. Wynik tego procesu po n -tym kroku jest zilustrowany na rys. 4.4. jako lewy kontekst-pudełko. Jednak proces może trwać dalej. W wyniku kolejnego kroku ze zdań wewnątrz kontekstu została usunięta informacja s (jest to jedynie symboliczne przedstawienie i nie należy odnosić tego np. do zmiennych), a na liście parametrów pojawił się dodatkowy parametr o nazwie *sit* i wartości s . Opisana czynność jest odwracalna, więc możliwa jest

dekontekstualizacja, czyli usunięcie parametru *sit* i wstawienie *s* do wnętrza kontekstu, lecz dotyczy to jedynie parametru jawnego.



Rys. 4.4 Metoda wstawiania i pobierania informacji zewnętrznej (za [BBG2000]).

Druga metoda, metoda przesuwania, polega na tym, że wybrany parametr nie znika, lecz zmienia się jego wartość. Na rys. 4.5 widoczna jest zmiana wartości *parametru t*. Jeśli rzeczywiście w dniu 1 stycznia 2010 roku padał śnieg (lub deszcz, co jest mało prawdopodobne), to w kontekście określonym tą datą prawdziwe jest zdanie „Dziś pada”, a w kontekście określonym datę o jeden dzień późniejszą zdanie „Wczoraj padało”.



Rys. 4.5 Metoda przesuwania informacji zewnętrznej (na podst. [BBG2000]).

Możliwa jest zmiana wartości innych parametrów, niekoniecznie o charakterze porządkującym. Na przykład zmiana może dotyczyć punktu widzenia różnych osób.

W [BBG2000] (także w [BBG2001]) opisano cechy kontekstów, które powinny być wspólne dla wszystkich proponowanych rozwiązań. Każdy z kontekstów zawiera fragment wiedzy o świecie, który da się scharakteryzować wg trzech *wymiarów*:

- zakres (ang. *partiality*) — jaką część świata, czy może raczej stanu rzeczy, kontekst opisuje,
- poziom szczegółowości (ang. *approximation*) — z jaką dokładnością dany problem jest ujęty (jakie szczegóły są uwzględnione, a jakie pominięte),
- perspektywa (ang. *perspective*) — z jakiego punktu widzenia dany problem jest opisany (przestrzennego, czasowego itp.).

Z kolei proces dowodzenia w ramach kontekstowej bazy wiedzy musi mieć dwie cechy: *lokalność* i *kompatybilność*. Pierwsza z cech oznacza, że podczas dowodzenia w ramach danego kontekstu mogą być brane pod uwagę tylko zdania w nim dostępne, tak jakby była to jedyna informacja o świecie, jaka jest do dyspozycji. Druga z cech mówi, że jako prawdziwa może zostać uznana jedynie ta formuła, która nie jest sprzeczna z przyjętymi zasadami dopuszczania wniosków wyprowadzonych w innych kontekstach. Autorzy opracowania [BBG2000] formułują dwie reguły dowodzenia kontekstowego. Każda z reguł

ma swoją wersję dotyczącą modelu oraz wersję dotyczącą dowodzenia. Sformułowanie tych reguł wymaga wprowadzenia kolejnych pojęć. Wiedza o danej dziedzinie jest reprezentowana przez aksjomatyczny system logiczny $Th = (L, Ak, R)$, gdzie L jest sformalizowanym językiem, Ak jest zbiorem aksjomatów wyrażonych w L , a R jest zbiorem reguł dowodzenia dla L . Wiedza kontekstowa jest z kolei zbiorem aksjomatycznych systemów logicznych $\{Th_i \mid 1 \leq i \leq n\}$ takich, że $Th_i = (L_i, Ak_i, R_i)$. Kontekstowa baza wiedzy zawiera konteksty $\kappa_1, \dots, \kappa_n$ i każdemu kontekstowi κ_i odpowiada system logiczny Th_i . Niech $Th(Ak_i)$ oznacza teorię rozumianą jako zbiór wszystkich formuł poprawnych w L_i niesprzecznych z aksjomatami Ak_i , domknięty ze względu na konsekwencje logiczne. Oczywiście $Th(Ak_i)$ jest podzbiorem L_i . Niech M_i oznacza zbiór wszystkich modeli dla języka L_i . Z kolei $M_{Th_i} \subseteq M_i$ jest zbiorem modeli spełniających teorię $Th(Ak_i)$.

Reguła 2.1a (lokalność — wersja semantyczna).

Zbiór modeli spełniających kontekst κ_i jest podzbiorem M_i .

Reguła 2.1b (lokalność — wersja dedukcyjna).

Formuła, która może być wyprowadzona w kontekście κ_i musi należeć do L_i .

Jeśli założy się monotoniczność kontekstu, to można tę regułę zapisać w silniejszej postaci:

Reguła 2.1a' (lokalność — wersja semantyczna)

Zbiór modeli spełniających kontekst κ_i jest podzbiorem M_{Th_i} .

Reguła 2.1b' (lokalność — wersja dedukcyjna)

Formuła, która może być wyprowadzona w kontekście κ_i musi należeć do $Th(Ak_i)$.

Kolejna reguła związana jest z uwzględnianiem informacji zewnętrznej. Informacje te (po odpowiednim przetłumaczeniu) mogą być postrzegane jako ograniczenia przestrzeni dostępnych modeli. Ta idea znajduje odzwierciedlenie w formalnym pojęciu ograniczenia strukturalnego. Niech SR będzie *ograniczeniem strukturalnym* między formułą φ_i w κ_i a formułą φ_j w κ_j . Model m_i dla języka L_i jest *kompatybilny* z modelem m_j dla języka L_j ze względu na SR wtedy, gdy spełnienie φ_i przez m_i pociąga za sobą spełnienie φ_j przez m_j . Model m_j dla języka L_j *spełnia ograniczenie strukturalne* SR z modelami dla L_i , gdy istnieje model m_i dla L_i kompatybilny z m_j (ze względu na SR).

W ramach systemu dowodzenia odpowiednikiem ograniczeń strukturalnych jest *strukturalne wynikanie*. Jest to takie wynikanie, które uwzględnia istnienie ograniczeń strukturalnych istniejących między formułami zawartymi w kontekście, w którym odbywa się dowodzenie, a formułami zawartymi w innych kontekstach danej bazy wiedzy. Niech symbol C_{SRi} oznacza zbiór zdań wyprowadzonych w kontekście κ_i dzięki strukturalnemu wynikaniu.

Reguła 2.2a (kompatybilność — wersja semantyczna)

Dany kontekst κ_i może być spełniony przez model m_i , jeżeli m_i spełnia wszystkie ograniczenia strukturalne z modelami innych kontekstów danej bazy wiedzy.

Reguła 2.2b (kompatybilność — wersja dedukcyjna)

Formuła, która może być wyprowadzona w kontekście κ_i , musi należeć do $Th(Ak_i \cup C_{SRi})$.

W kolejnych punktach rozdziału zostaną opisane przykładowe systemy logiczne, które uwzględniają istnienie kontekstów.

4.1.2 Logika zaimków wskazujących

Logika zaimków wskazujących (ang. *the logic of demonstratives — LD*) została zaproponowana przez Davida Kaplana w [K1978]. Zaimki wskazujące występują w języku

naturalnym i odnoszą się do znanych z kontekstu (wypowiedzi) pojęć lub ich cech. Przykładowymi zaimkami wskazującymi są: *tu*, *tam*, *teraz*, *ten*, *tamten*, itp. Autor włącza również do swoich rozważań zaimki osobowe: *ja*, *ty*, *on*. Motywacją do wprowadzenia własnego rozwiązania kontekstowego było spostrzeżenie, że dotychczasowe sposoby wartościowania zmiennych symulujących zaimki miały się z intuicją. Przykładem jest wartościowanie zdania $p_0 = \text{„(Ja) jestem tu teraz”}$ i zdania $p_1 = \Box \text{„(Ja) jestem tu teraz”}$ ¹⁴. Otóż zdanie p_0 w większości światów możliwych ma wartość logiczną fałszu, gdyż osoba je wypowiadająca mogłaby się w danej chwili znajdować gdzie indziej, w innym niż *ten* świecie możliwym. Dlatego też zdanie p_1 nie jest prawdziwe. A jednak, zgodnie z intuicją, zdanie p_0 jest zawsze prawdziwe. Gdziekolwiek i kiedykolwiek ktokolwiek je wypowie, zawsze będzie ono odzwierciedlało prawdę. Kaplan stwierdza, że przyczyną takiego odejścia formalizmu od języka naturalnego jest prawdopodobnie fakt, że wraz ze zmianą kontekstu zmienia się nie tylko wartościowanie zdania, lecz także jego *zawartość* (czyli to, co ma być wartościowane), którą wstępnie definiuje jako funkcję ze zbioru światów możliwych do zbioru agentów (osobników wypowiadających zdanie). Istnieje jakaś relacja, nazywa ją *charakterem* zdania, która wiąże kontekst z tą zawartością. Relacja ta działa w ten sposób, że nie wszystkie światy możliwe są brane pod uwagę podczas wartościowania, lecz tylko te, które są *właściwe* dla danego kontekstu. Relacja ta powoduje, że w obrębie właściwych światów możliwych zdanie p_1 staje się prawdziwe.

Kierując się opisaną motywacją, Kaplan proponuje logikę zaimków wskazujących. Bazuje ona na logice pierwszego rzędu, logice modalnej i temporalnej. Ma bogatą (i skomplikowaną) syntaktykę. Jeśli chodzi o logikę pierwszego rzędu, to wzbogaca ją o dwie kategorie gramatyczne: reprezentujące osoby¹⁵ (ang. *individual sort*) i pozycje (ang. *position sort*). Wprowadzenie tych dwóch kategorii pociąga za sobą istnienie dwóch typów termów — *i*-termy reprezentujące osoby oraz *p*-termy reprezentujące pozycje. Zarówno zmienne, jak i funkcje muszą być określonego typu. Definicje predykatów i funkcji muszą określać typ argumentów. Z logiki modalnej zapożyczone zostały operatory modalne, a z logiki temporalnej operatory *F* (będzie miało miejsce, że), *P* (miało miejsce, że) i *G* (miało miejsce dzień wcześniej, że). Oczywiście syntaktyka LD definiuje także zaimki. Są to stałe *I* i *here*, dwie funkcje jednoargumentowe *dthat_i* i *dthat_p* oraz jednoargumentowe operatory zdaniowe *N* (teraz ma miejsce, że), *A* (w danym świecie możliwym ma miejsce, że) i *Y* (wczoraj miało miejsce, że). Funkcje *dthat_i* i *dthat_p* przyjmują jako argument odpowiednio *i*-term i *p*-term i zwracają term tego samego typu. Operatory zdaniowe działają na formułach. Równie bogata jak syntaktyka jest semantyka logiki zaimków wskazujących. Interpretacją jest tu struktura $\mathcal{U} = (S_C, S_W, S_U, S_P, S_T, f)$, gdzie:

- 1) S_C jest niepustym zbiorem kontekstów; jeżeli $c \in S_C$, to:
 - a) $c_A \in S_U$ jest agentem c ,
 - b) $c_T \in S_T$ jest czasem c ,
 - c) $c_P \in S_P$ jest pozycją c ,
 - d) $c_W \in S_W$ jest światem c ;
- 2) S_W jest niepustym zbiorem światów możliwych,
- 3) S_U jest niepustym zbiorem obiektów reprezentujących osoby,

¹⁴ Zdanie z operatorem modalnym „konieczne, że”. Można więc to zdanie przetłumaczyć na: „To konieczne, że ja jestem tu teraz”, ale, jak wynika z dalszych rozważań, takie tłumaczenie nie jest zawsze poprawne.

¹⁵ Kaplan stosuje słowo *individual*, mając na myśli kogoś, kto wypowiada zdanie. Przetłumaczenie tego słowa na *osobnik* stworzyłoby konflikt z nazwą obiektu w logice opisowej. Z kolei słowa *agent* Kaplan używa do oznaczenia osoby związanej z kontekstem — jest to *agent kontekstu*.

- 4) S_P jest niepustym zbiorem pozycji, wspólnym dla wszystkich światów możliwych,
- 5) S_T jest niepustym zbiorem liczb całkowitych reprezentujących punkty czasowe, wspólnych dla wszystkich światów możliwych,
- 6) j jest przyporządkowaniem interpretacyjnym.

Nie wchodząc zbyt głęboko w opis tej logiki, warto może naszkicować sposób, w jaki działa interpretacja. Dla ustalonej struktury \mathcal{U} wartościowanie formuł i termów zawsze zachodzi z uwzględnieniem kontekstu, funkcji wartościującej zmienne, czasu i świata możliwego.

Zapis $\models_{cvtw} \varphi$ oznacza, że formuła φ jest wartościowana w kontekście c i dla funkcji wartościującej zmienne v ma wartość logiczną prawdy z uwzględnieniem czasu t i świata w .

Zapis $|x|_{cvtw}$ oznacza przypisanie zmiennej wartości w kontekście c przez funkcję wartościującą zmienne v z uwzględnieniem czasu t i świata w .

Dla wymienionych wyżej zaimków semantyka została zdefiniowana w sposób następujący:

- 1) $|I|_{cvtw} = c_A$,
- 2) $|here|_{cvtw} = c_P$,
- 3) $|dthat(x)|_{cvtw} = |x|_{cvc_tcw}$,
- 4) $\models_{cvtw} N\varphi$ wtwg $\models_{cvc_{tw}} \varphi$,
- 5) $\models_{cvtw} A\varphi$ wtwg $\models_{cvt_{cw}} \varphi$,
- 6) $\models_{cvtw} Y\varphi$ wtwg $\models_{cv(c_T-1)w} \varphi$.

Ogólnie mówiąc, wyrażenia zawierające zaimki znaczą co innego w różnych kontekstach. To znaczenie, które autor nazywa zawartością, dzięki przedstawionej semantyce można sformalizować w sposób następujący. *Zawartością formuły φ* dla danego kontekstu c i funkcji wartościującej zmienne v (przy ustalonej interpretacji \mathcal{U}), oznaczaną $\{\varphi\}_{cv}$, jest funkcja, która parze $t \in S_T$ i $w \in S_W$ przypisuje wartość logiczną prawdy, gdy $\models_{cvtw} \varphi$ i wartość fałszu w przeciwnym przypadku. Można więc napisać: $\models_{cvtw} \varphi$ wtwg $\{\varphi\}_{cv}(t, w) = 1$. Mówiąc mniej formalnie, formuła φ w kontekście c jest prawdziwa z uwzględnieniem t i w wtedy, gdy treść, którą ta formuła reprezentuje po umieszczeniu jej w kontekście c , jest prawdziwa w czasie t i gdy aktualnym światem jest w .

Z kolei *charakterem formuły φ* , zapisywanym $\{\varphi\}$, jest funkcja, która parze $c \in S_C$ i v (funkcji wartościującej zmienne) przypisuje zawartość formuły φ . Można więc zapisać: $\{\varphi\}(cv) = \{\varphi\}_{cv}$.

Formuła φ ma *stabilny charakter*, jeżeli we wszystkich kontekstach ma taką samą zawartość, czyli $\forall c' \neq c \in S_C (\{\varphi\}_{cv} = \{\varphi\}_{c'v})$.

Wadą przedstawionego tu systemu jest duży stopień komplikacji wynikający z tego, że autor dążył do odzwierciedlenia w jego wnętrzu przyczyn powstawania kontekstów. Systemy opisane w kolejnych rozdziałach wychodzą z innego założenia. To użytkownik ma decydować o tym, dlaczego i w jakim celu tworzony jest dany kontekst. Informacja o przyczynie i celu nie jest też w żaden sposób reprezentowana. Natomiast system dostarcza metod, aby zdefiniować sposób, w jaki informacja umieszczona w jednym kontekście wpływa na wnioskowanie prowadzone wewnątrz innego kontekstu. Takie ujęcie problemu wpłynęło na znaczne uproszczenie tych systemów.

4.1.3 Kontekstowa logika zdań (PLC)

Pomysł wprowadzenia kontekstowej logiki zdań PLC zrodził się podczas prac nad opisem wiedzy powszechnej prowadzonych w ramach projektu CYC, który dziś dysponuje największą bazą wiedzy, jaką kiedykolwiek stworzono. Gdy baza ta osiągnęła wielkość rzędu 100 000 zdań, niezbędne stało się podzielenie jej na *konteksty*. Logika PLC została zaprezentowana przez McCarthy'ego na konferencji IJCAI w [M1993]. Pełny jej opis formalny znaleźć można w [G1992] i [BBM1994].

PLC rozszerza logikę pierwszego rzędu o pojęcie kontekstu i właściwości z nim związane. Po pierwsze, przyjmuje się, że każde wypowiedziane zdanie znajduje się w jakimś kontekście. Nie istnieje więc po prostu zdanie p , lecz $k:p$, co oznacza, że zdanie p zostało wypowiedziane w kontekście k . Nawet jeśli kontekst k zostanie pominięty w zapisie, to mimo to musi zostać założony jakiś domyślny kontekst dyskursu. Po drugie, wartość logiczna danego zdania, niezależnie od tego, gdzie zostało wypowiedziane, musi być określona w jakimś kontekście. Może bowiem być tak, że dane zdanie jest prawdziwe w jednym kontekście, a fałszywe lub pozbawione znaczenia w innym. Na przykład, zdanie „Król Francji jest łysy” (przykład zaczerpnięty z [McCB1994]) ma inną wartość logiczną w kontekście XIII wieku, inną w kontekście Rewolucji Francuskiej, a w dniu dzisiejszym jest pozbawione wartości logicznej, gdyż Francja nie jest już królestwem. Dla wyrażenia wartości logicznej zdania stosuje się *predykat modalny* ist . Można więc powiedzieć $ist(k, p)$, co będzie oznaczało, że zdanie p jest prawdziwe w kontekście k . Oczywiście, należy to powiedzieć w jakimś kontekście k_0 .

Jak można zauważyć, konteksty są równouprawnionymi obiektami dziedziny, o których można się wypowiadać w zdaniach. Konteksty nie mogą być w pełni zdefiniowane¹⁶. Można je sobie wyobrazić jako zbiory zdań zdefiniowane innymi zdaniami opisującymi założenia tworzące tło kontekstu. Gdyby liczba tych założeń była skończona, można byłoby dokonać dekontekstualizacji kontekstu poprzez dołączenie zdań opisujących te założenia do zdań znajdujących się wewnątrz kontekstu. Jednakże zbiór zdań opisujących założenia jest zbiorem nieskończonym — zawsze da się go rozbudować o kolejne zdania coraz bardziej precyzyjnie opisujące tło. Tak więc zakłada się, że definicja kontekstu jest niejawna, że zbiór zdań opisujących jego tło jest znany i milcząco akceptowany przez osoby zeń korzystające.

4.1.3.1 Syntaktyka PLC

Syntaktyka PLC uwzględnia istnienie kontekstów i predykatu modalnego ist . Niech symbol P oznacza zbiór zdań atomowych, a K zbiór kontekstów. Zbiór poprawnie zbudowanych formuł zdaniowych W może być wyprowadzony z gramatyki zapisanej w notacji BNF w następujący sposób:

- 1) $form ::= p$ dla dowolnego $p \in P$
- 2) $form ::= \neg form$
- 3) $form ::= form \vee form$
- 4) $form ::= form \wedge form$
- 5) $form ::= form \rightarrow form$

¹⁶ W publikacjach o sztucznej inteligencji pojawia się czasem pojęcie *rich object*, czyli obiektu, który nie da się w pełni opisać, wprowadzone w [MH1969].

6) $form ::= form \leftrightarrow form$

7) $form ::= \text{ist}(k, form)$ dla dowolnego $k \in K$

Należy pamiętać o tym, że jest to zapis uproszczony, zakładający istnienie domyślnego kontekstu. Właściwie każde zdanie powinno być poprzedzone symbolem kontekstu, np.:

$$k_1:p \vee k_2:q \leftrightarrow \neg k_1:p \rightarrow k_2:q$$

4.1.3.2 Semantyka

Początkowo ([G1992]) kontekst w PLC był interpretowany po prostu jako zbiór funkcji wartościujących zdania. Formuła $\text{ist}(k, p)$ była prawdziwa, jeśli zdanie p było prawdziwe we wszystkich wartościowaniach związanych z kontekstem k . Semantyka była definiowana jako zbiór odwzorowań między kontekstami a tymi zbiorami wartościowań. Jednakże taka semantyka okazała się niewystarczająca, aby zawrzeć niektóre właściwości kontekstów. Po pierwsze, natura danego konceptu może być zależna od innych kontekstów. Przykładowo, kontekst wyścigów samochodowych z lat 50. XX wieku różni się od wyścigów współczesnych. Formuła $\text{ist}(k_2, p)$ wypowiedziana w kontekście k_1 oznacza, że z punktu widzenia k_1 zdanie p jest prawdziwe w k_2 . Tak więc wartościowanie nie powinno być cechą kontekstu, lecz ciągu kontekstów (k_0, k_1, \dots, k_n) . Taki ciąg kontekstów będzie oznaczany symbolem κ , a zbiór wszystkich możliwych skończonych ciągów symbolem \mathcal{K} .

Po drugie, nie ma potrzeby, aby wszystkie zdania miały wartość logiczną w każdym kontekście. Zdania sensowne w kontekście historii detektywa Sherlocka Holmesa, opowiedzianej przez Artura Conan-Doyle'a, nie są sensowne w kontekście teorii grawitacji. Wynika z tego, że wartościowanie zdań w kontekstach powinno być funkcją częściową.

Definicja modelu dla PLC jest następująca:

Definicja 4.1 (model PLC)

Model \mathcal{M} dla PLC jest częściową funkcją, która odwzorowuje każdy ciąg kontekstów w zbiór częściowych wartościowań zbioru P :

$$\mathcal{M} \in \mathcal{KP}(\{v \in P \mid \text{Fnc}(v) \text{ oraz } \text{dom}(v) \subseteq P\})$$

(gdzie $\mathcal{P}(A)$ to zbiór potęgowy zbioru A) oraz dla $\forall \kappa \in \mathcal{K}$ i $v_1, v_2 \in \mathcal{M}(\kappa)$ prawdziwa jest równość $\text{dom}(v_1) = \text{dom}(v_2)$. ■

Pośrednio, z powodu tego, że v jest funkcją częściową, model \mathcal{M} definiuje relację, która wiąże dany podzbiór zdań z danym kontekstem. Relacja nazywa się *słownikiem* i jest oznaczana symbolem $\mathcal{S}(\mathcal{M})$. Formalnie ujmując, relacja ta jest podzbiorem iloczynu kartezjańskiego zbioru ciągów kontekstów \mathcal{K} i zbioru zdań atomowych P : $\mathcal{S} \subseteq \mathcal{K} \times P$. Para $(\kappa, p) \in \mathcal{S}(\mathcal{M})$ wtedy i tylko wtedy, gdy zdefiniowany jest model $\mathcal{M}(\kappa)$ dla ciągu kontekstów κ i dla wszystkich $v \in \mathcal{M}(\kappa)$ zdefiniowane jest $v(p)$.

Ograniczoność słownika danego kontekstu powoduje, że nie wszystkim formułom można przypisać wartość logiczną. Aby zdefiniować spełnialność i prawdziwość formuł, należy najpierw wprowadzić pojęcie słownika formuły. Każda formuła wypowiedziana w danym kontekście wyznacza swój własny słownik, oznaczany jako $\mathcal{S}(\kappa, \varphi)$, który jest najmniejszym podzbiorem słownika $\mathcal{S}(\mathcal{M})$ niezbędnym do zbudowania formuły φ . Słownik formuły tworzy się rekurencyjnie, korzystając z następujących reguł:

$$\mathcal{S}(\kappa, p) = \{(\kappa, p)\}$$

$$\begin{aligned} S(\kappa, \neg\varphi) &= S(\kappa, \varphi) \\ S(\kappa, \varphi \rightarrow \psi) &= S(\kappa, \varphi) \cup S(\kappa, \psi) \\ S(\kappa_1, \text{ist}(\kappa_2, \varphi)) &= S(\kappa_2, \varphi) \end{aligned}$$

Mając określone pojęcie słownika formuły, można już zdefiniować spełnialność i prawdziwość formuł w systemie PLC.

Definicja 4.2 (spełnianie, spełnialność i prawdziwość)

Niech φ i \mathcal{M} będą odpowiednio formułą zdaniową i modelem powiązanymi zależnością $S(\kappa, \varphi) \subseteq S(\mathcal{M})$. φ jest spełniona w \mathcal{M} poprzez przypisanie $v \in \mathcal{M}(\kappa)$, co zapisujemy $\mathcal{M}, v \models_{\kappa} \varphi$, zgodnie z następującymi regułami:

- 1) $\mathcal{M}, v \models_{\kappa} p$ wtedy i tylko wtedy, gdy $v(p) = 1$;
- 2) $\mathcal{M}, v \models_{\kappa} \neg\varphi$ wtedy i tylko wtedy, gdy $\mathcal{M}, v \not\models_{\kappa} \varphi$;
- 3) $\mathcal{M}, v \models_{\kappa} \varphi \rightarrow \psi$ wtedy i tylko wtedy, gdy $\mathcal{M}, v \models_{\kappa} \psi$ lub $\mathcal{M}, v \not\models_{\kappa} \varphi$;
- 4) $\mathcal{M}, v \models_{\kappa} \text{ist}(k, \varphi)$ wtedy i tylko wtedy, gdy dla wszystkich wartościowań formuł $v' \in \mathcal{M}(\kappa.k)$ prawdziwe jest $\mathcal{M}, v' \models_{\kappa.k} \varphi$;
- 5) $\mathcal{M} \models_{\kappa} \varphi$ wtedy i tylko wtedy, gdy dla wszystkich wartościowań $v \in \mathcal{M}(\kappa)$ prawdziwe jest $\mathcal{M}, v \models_{\kappa} \varphi$;
- 6) $\models_{\kappa} \varphi$ wtedy i tylko wtedy, gdy dla wszystkich modeli \mathcal{M} takich, że $S(\kappa, \varphi) \subseteq S(\mathcal{M})$ prawdziwe jest $\mathcal{M} \models_{\kappa} \varphi$;

Formuła φ jest *prawdziwa* w ciągu kontekstów κ , jeśli $\models_{\kappa} \varphi$, φ jest *spełnialna* w ciągu kontekstów κ , jeśli istnieje model \mathcal{M} taki, że $\mathcal{M} \models_{\kappa} \varphi$. Zbiór formuł S jest spełnialny w kontekście κ , jeśli istnieje taki model \mathcal{M} , że $\mathcal{M} \models_{\kappa} \varphi$ dla wszystkich $\varphi \in S$. ■

4.1.3.3 System dowodzenia

Prezentując logikę kontekstową PLC, autorzy skupiają się na hilbertowskim systemie dowodzenia. Istniejące w klasycznym rachunku zdań reguły dowodzenia zostały uzupełnione o elementy uwzględniające istnienie kontekstów. Dodane reguły dowodzenia opracowano w celu wsparcia specyficznej techniki dowodzenia polegającej na wchodzeniu (reguła Enter) i wychodzeniu (reguła Exit) z kontekstu. Przeprowadzając dowodzenie, należy kierować się zasadą, że w danym momencie uwzględniamy jedynie formuły spełnialne w bieżącym kontekście. O tym, czy w celu dalszego dowodzenia należy przejść do innego kontekstu, decyduje zbiór *aksjomatów przenoszących* (ang. *lifting axioms*), tworzących powiązania między kontekstami. Po przejściu do innego kontekstu dowodzenie jest kontynuowane w kierunku wyprowadzenia formuły docelowej jedynie na podstawie formuł prawdziwych w tym kontekście. W końcu następuje powrót do kontekstu poprzedniego. W dalszej części rozprawy zaprezentowany zostanie przykład ilustrujący taki sposób przeprowadzania dowodu.

W przedstawionym zbiorze reguł dowodzenia wykorzystywany jest znak symbolizujący relację wyprowadzania, która jest definiowana w następujący sposób.

Definicja 4.3 (wyprowadzalność)

Formuła zdaniowa φ jest *wyprowadzalna w ciągu kontekstów* κ (co jest zapisywane $\vdash_{\kappa} \varphi$) wtedy i tylko wtedy, gdy φ jest tautologią w κ lub wynika z innych formuł wyprowadzalnych

przy użyciu jednej z reguł dowodzenia, czyli wtedy i tylko wtedy, gdy istnieje ciąg $(\vdash_{\kappa_1} \varphi_1, \dots, \vdash_{\kappa_n} \varphi_n)$ taki, że $\kappa_n = \kappa$ i $\varphi_n = \varphi$, i dla każdego $i \in [2, n]$ element ciągu $\vdash_{\kappa_i} \varphi_i$ albo jest tautologią, albo jest wyprowadzony z elementu wcześniejszego przy użyciu jednej z reguł dowodzenia. W przypadku istnienia aksjomatów, formuła φ jest *wyprowadzalna ze zbioru aksjomatów T* w ciągu kontekstów κ (co zapisujemy $T \vdash_{\kappa} \varphi$) wtedy i tylko wtedy, gdy istnieją formuły $\varphi_1, \dots, \varphi_n \in T$ takie, że $\vdash_{\kappa} (\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow \varphi$. ■

Oto przykładowe reguły dowodzenia:

$$\vdash_{\kappa} \varphi \quad \text{jeżeli } \varphi \text{ jest tautologią} \quad (\text{PL})$$

$$\frac{\vdash_{\kappa} \varphi, \quad \vdash_{\kappa} \varphi \rightarrow \psi}{\vdash_{\kappa} \psi} \quad (\text{MP})$$

$$\frac{\vdash_{\kappa} \varphi, \quad \vdash_{\kappa} \psi}{\vdash_{\kappa} \varphi \rightarrow \psi} \quad (\text{II})$$

$$\vdash_{\kappa} \text{ist}(k, \varphi \rightarrow \psi) \rightarrow \text{ist}(k, \varphi) \rightarrow \text{ist}(k, \psi) \quad (\text{K})$$

$$\vdash_{\kappa} \text{ist}(k_1, \text{ist}(k_2, \varphi) \vee \psi) \rightarrow \text{ist}(k_1, \text{ist}(k_2, \varphi)) \vee \text{ist}(k_1, \psi) \quad (\Delta)$$

$$\frac{\vdash_{\kappa} \text{ist}(k, \varphi)}{\vdash_{\kappa.k} \varphi} \quad (\text{Enter})$$

$$\frac{\vdash_{\kappa.k} \varphi}{\vdash_{\kappa} \text{ist}(k, \varphi)} \quad (\text{Exit})$$

Reguły (PL), (MP) i (II) gwarantują, że wszystkie prawdziwe formuły klasycznego rachunku zdań są również prawdziwe w każdym ciągu kontekstów i że każdy ciąg kontekstów jest domknięty ze względu na klasyczne reguły dowodzenia. Kolejne reguły opisują właściwości dowodzenia kontekstowego. Reguła (K) gwarantuje, że każdy kontekst jest domknięty ze względu na wnioski. Właściwość, która nazywana jest „kontekstową wszechwiedzą” (ang. *contextual omniscience*), jest opisana przez regułę (Δ). Wynika z niej, że każdy kontekst „wie”, co jest prawdą w każdym innym kontekście. Tak więc, chociaż kontekst nie musi mieć kompletnej informacji o tym, co jest prawdą w całym opisywanym świecie, to dysponuje kompletną informacją o tym, jak ten świat jest „widziany” przez inny kontekst. Reguły (Enter) i (Exit) pozwalają na wchodzenie i wychodzenie z kontekstów. Łatwo zauważyć, że reguła (Enter) jest odwróceniem reguły (Exit). Po kilkukrotnym zagłębieniu się przy wykorzystaniu reguły (Enter) w kolejne „warstwy” kontekstów następuje, dzięki regule (Exit), powrót do kontekstu początkowego. Czasem jednak następuje *wykraczanie* (ang. *transcending*) poza tak utworzoną *ścieżkę* do tzw. *kontekstu zewnętrznego* (ang. *outer context*). Rola takiego kontekstu zostanie opisana następnym punkcie.

Przedstawiony zbiór reguł dowodzenia jest przykładowy w tym sensie, że może być uzupełniany w celu uzyskania różnych poziomów ekspresji. Na przykład często dołączana jest reguła:

$$\vdash_{\kappa} \text{ist}(k_2, \text{ist}(k_1, \varphi)) \rightarrow \text{ist}(k_1, \varphi) \quad (\text{Flat})$$

Reguła ta mówi, że każdy kontekst wygląda tak samo, niezależnie od tego, z jakiego kontekstu i w jakim ciągu kontekstów jest widziany. Upraszcza ona semantykę, gdyż powoduje, że sposób wartościowania zdań nie zależy już od wyboru ciągu kontekstów, lecz od pojedynczego kontekstu.

Ze względu na to, że bazy wiedzy budowane są w języku predykatów, zbiór reguł jest uzupełniany o reguły pozwalające korzystać ze zdań zawierających kwantyfikatory, predykaty, nazwy obiektów i zmienne.

4.1.3.4 Struktura kontekstów

Rozważając różne podejścia do kontekstualizacji baz wiedzy, warto zwrócić uwagę na sposób, w jaki użytkownik systemu PLC tworzy strukturę kontekstów. Problem można rozpatrywać z dwóch perspektyw: gdy strukturą nazwie się statyczne związki między kontekstami utworzone na etapie projektowania bazy wiedzy lub gdy za strukturę uzna się zbiór kontekstów uczestniczących w procesie dowodzenia, dynamicznie utworzony podczas tego procesu.

W pierwszym przypadku użytkownik korzysta z tzw. *aksjomatów przenoszących* (ang. *lifting axioms*). Z punktu widzenia syntaktyki, aksjomaty te mogą być definiowane w różnych kontekstach. Z punktu widzenia semantycznego, muszą oczywiście być rozmieszczone w taki sposób, aby wspomóc proces dowodzenia. Najprostszym sposobem powiązania dwóch kontekstów jest zaimportowanie wszystkich formuł zdefiniowanych w pierwszym z nich do drugiego. Można to zrobić wewnątrz kontekstu importującego:

$$k_2: \forall \varphi (\text{ist}(k_1, \varphi) \rightarrow \varphi)$$

lub w kontekście zewnętrznym:

$$k_0: \forall \varphi (\text{ist}(k_1, \varphi) \rightarrow \text{ist}(k_2, \varphi))$$

Zdefiniowanie aksjomatów przenoszących w kontekście zewnętrznym zmienia sposób dowodzenia. Konieczne jest wtedy wykorzystanie mechanizmu wykraczania poza ścieżkę z użyciem reguł (Enter) i (Exit). Komplikuje to proces tworzenia dowodu, gdyż powoduje konieczność uwzględnienia w trakcie dowodzenia większej liczby kontekstów. To oczywiście zwiększa złożoność pamięciową problemu. Z drugiej strony upraszcza to zawartość samych kontekstów, co poprawia wydajność dowodzenia wewnątrz danego kontekstu.

Jeśli w ramach jednego kontekstu istnieje potrzeba skorzystania jedynie z części wiedzy zawartej w drugim, należy zdefiniować precyzyjne aksjomaty przenoszące, uwzględniając tylko te formuły, które są interesujące. W kontekstowej logice predykatów polega to na definiowaniu odpowiedniości predykatów. I tak na przykład w następujący sposób możemy powiązać kontekst prawny ustawy o VAT z kontekstem informacji motoryzacyjnych:

$$k_0: \forall x (\text{ist}(\text{VAT}, \text{SamochódCiężarowy}(x)) \rightarrow \text{ist}(\text{motoInfo}, \\ \text{PojazdCiężarowy}(x) \vee \text{PojazdOsobowy}(x) \wedge \text{MaZamontowane}(x, \text{kratka})))$$

4.1.3.5 Przykład

Podany poniżej przykład ma na celu zilustrowanie sposobu tworzenia dowodu w systemie PLC. Ponieważ jest on wyrażony w klasycznym rachunku predykatów, należy do zbioru reguł dowodzenia dodać nowe, znane jako reguły konkretyzacji (*universal instantiation* — UI) oraz generalizacji uniwersalnej (*universal generalization* — UG):

$$\frac{\vdash_{\kappa} \forall x \varphi(x)}{\vdash_{\kappa} \varphi(x)} \quad (\text{UI})$$

$$\frac{\vdash_{\kappa} \varphi \rightarrow \psi(x)}{\vdash_{\kappa} \forall x \psi(x)} \quad (x \text{ nie jest zmienną wolną w formule } \varphi) \quad (\text{UG})$$

Zostały one dostosowane do systemu PLC w ten sposób, aby uwzględnić fakt, że dowodzenie odbywa się w ciągu kontekstów κ . Oznaczenia $\varphi(x)$ i $\psi(x)$ zostały tu użyte jako formuły w rachunku predykatów zależne od zmiennej x .

Przykład 4.2 (projekty medyczne w systemie PLC)

Założmy, że istnieje kontekst o nazwie *projekty*, który opisuje cechy projektów naukowych oraz zależności między różnymi ich rodzajami. Kontekst ten zawiera następujące aksjomaty:

$$\text{projekty: } \forall x (\text{Projekt_dot_schorzeń_genetycznych}(x) \leftrightarrow (\text{Projekt}(x) \wedge \exists y (\text{dotyczy}(x, y) \wedge \text{Schorzenie_genetyczne}(y))))), \quad (4.5)$$

$$\text{projekty: } \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \leftrightarrow (\text{ProjektUE}(x) \wedge \exists y (\text{dotyczy}(x, y) \wedge \text{Mukowiscydoza}(y))))), \quad (4.6)$$

$$\text{projekty: } \forall x (\text{ProjektUE}(x) \rightarrow \text{Projekt}(x)), \quad (4.7)$$

$$\text{projekty: } \forall x (\exists y \text{ dotyczy}(x, y) \rightarrow \text{Projekt}(x)). \quad (4.8)$$

Należy udowodnić formułę:

$$k_0: \text{ist}(\text{projekty}, \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \rightarrow \text{Projekt_dot_schorzeń_genetycznych}(x))). \quad (4.9)$$

Ponieważ formuła ta nie wynika z kontekstu *projekty*, należy posłużyć się dodatkowym kontekstem *schorzenia*, w którym umieszczone są m.in. następujące aksjomaty:

$$\text{schorzenia: } \forall x (\text{Zwłóknienie_genet}(x) \leftrightarrow (\text{Zwłóknienie}(x) \wedge \exists y (\text{spowodowane_przez}(x, y) \wedge \text{Przyczyny_genet}(y))))), \quad (4.10)$$

$$\text{schorzenia: } \forall x (\text{Mukowiscydoza}(x) \leftrightarrow (\text{Zwłóknienie_genet}(x) \wedge \exists y (\text{zlokalizowane_w}(x, y) \wedge \text{Trzustka}(y))))). \quad (4.11)$$

Aby można było wykorzystać wiedzę z dwóch kontekstów, należy stworzyć dodatkowe aksjomaty przenoszące:

$$k_0: \forall x (\text{ist}(\text{projekty}, \text{Mukowiscydoza}(x)) \leftrightarrow \text{ist}(\text{schorzenia}, \text{Mukowiscydoza}(x))), \quad (4.12)$$

$$k_0: \forall x (\text{ist}(\text{projekty}, \text{Schorzenie_genetyczne}(x)) \leftrightarrow \text{ist}(\text{schorzenia}, \exists y (\text{spowodowane_przez}(x, y) \wedge \text{Przyczyny_genet}(y)))). \quad (4.13)$$

Dowód:

Dowód zaczynamy od formuły utworzonej dla dowolnego, lecz ustalonego x :

$$\text{projekty: ProjektUE_dot_mukowiscydozy}(x). \quad (4.14)$$

Ze zdania tego oraz ze względu na zdanie (4.6) prawdziwa jest także formuła:

$$\text{projekty: ProjektUE}(x) \quad (4.15)$$

oraz formuła:

$$\text{projekty: } \exists y (\text{dotyczy}(x, y) \wedge \text{Mukowiscydoza}(y)). \quad (4.16)$$

Po opuszczeniu kwantyfikatora i opuszczeniu pierwszego członu koniunkcji otrzymujemy:

$$\text{projekty: Mukowiscydoza}(y) \quad (4.17)$$

Teraz, korzystając z reguły (Exit), wchodzimy do kontekstu k_0 :

$$k_0: \text{ist}(\text{projekty}, \text{Mukowiscydoza}(y)). \quad (4.18)$$

Stąd oraz z aksjomaty przenoszącego (4.12):

$$k_0: \text{ist}(\text{schorzenia}, \text{Mukowiscydoza}(y)). \quad (4.19)$$

Następnie, dzięki regule (Enter) wchodzimy do kontekstu *schorzenia*:

$$\text{schorzenia: Mukowiscydoza}(y). \quad (4.20)$$

W ramach tego kontekstu wnioskujemy na podstawie (4.11), że prawdziwe jest zdanie:

$$\text{schorzenia: Zwłóknienie_genet}(y), \quad (4.21)$$

a na podstawie (4.10), że prawdziwe jest:

$$\text{schorzenia: } \exists z (\text{spowodowane_przez}(y, z) \wedge \text{Przyczyny_genet}(z))). \quad (4.22)$$

Znów korzystamy z reguły (Exit) i wracamy do kontekstu k_0 :

$$k_0: \text{ist}(\text{schorzenia}, \exists z (\text{spowodowane_przez}(y, z) \wedge \text{Przyczyny_genet}(z))). \quad (4.23)$$

Z powyższego zdania oraz z aksjomatu przenoszącego (4.13) otrzymujemy:

$$k_0: \text{ist}(\text{projekty}, \text{Schorzenie_genetyczne}(y)). \quad (4.24)$$

Ponownie korzystając z reguły (Enter) wchodzimy do kontekstu *projekty*:

$$\text{projekty: Schorzenie_genetyczne}(y). \quad (4.25)$$

Z powyższego zdania oraz z (4.7), (4.15) i (4.16) otrzymujemy:

$$\text{projekty: Projekt}(x) \wedge \exists y (\text{dotyczy}(x, y) \wedge \text{Schorzenie_genetyczne}(y)) \quad (4.26)$$

i dzięki aksjomatowi (4.5):

$$\text{projekty: Projekt_dot_schorzeń_genetycznych}(x). \quad (4.27)$$

Po zastosowaniu do (4.14) i (4.27) reguły wprowadzania implikacji (II) mamy:

$$\text{projekty: ProjektUE_dot_mukowiscydozy}(x) \rightarrow \text{Projekt_dot_schorzeń_genetycznych}(x), \quad (4.28)$$

a po zastosowaniu reguły generalizacji uniwersalnej (UG):

$$\text{projekty: } \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \rightarrow \text{Projekt_dot_schorzeń_genetycznych}(x)). \quad (4.29)$$

Następnie znów korzystamy z reguły (Exit), wracamy do kontekstu k_0 i otrzymujemy zdanie, które należało udowodnić:

$$k_0: \text{ist}(\text{projekty}, \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \rightarrow \text{Projekt_dot_schorzeń_genetycznych}(x))). \quad (4.30)$$

QED ■

W powyższym dowodzie dwukrotnie nastąpiło przejście z kontekstu do kontekstu na podstawie reguły (Enter) (w (4.20) i (4.25)) i trzykrotnie powrót do kontekstu k_0 na podstawie reguły (Exit) (w (4.18), (4.23) i (4.30)). Charakterystyczne jest to, że wewnątrz danego kontekstu dowód został przeprowadzony tylko na podstawie zdań, które były wcześniej

w nim zawarte. Można zauważyć, że z poziomu kontekstu k_0 „widoczne” są pozostałe konteksty konieczne do przeprowadzenia dowodu. W zademonstrowanym przykładzie kontekst k_0 został także wykorzystany do zdefiniowania aksjomatów przenoszących (4.12) i (4.13) opisujących zależność między formułami z kontekstów *projekty* i *schorzenia*. Aksjomaty przenoszące są częścią kontekstowej bazy wiedzy i stanowią o jej strukturze, rozumianej jako relacje między kontekstami. Uzależniają one prawdziwość zdań wypowiedzianych w jednym kontekście od zdań wypowiedzianych w innym.

4.1.4 Semantyka modeli lokalnych/systemy wielokontekstowe (LMS/MCS)

Nazwą *systemy wielokontekstowe* (ang. *MultiContext Systems — MCS*) [GS1994] została objęta klasa systemów dowodzenia dysponująca możliwością przenoszenia wniosków między dowodami przeprowadzanymi wewnątrz odrębnych kontekstów. W [GG2001] wprowadzona została *semantyka modelu lokalnego* (ang. *Local Models Semantics — LMS*), która na gruncie teorii modeli pozwala opisać semantykę MCS. Obie konstrukcje teoretyczne tworzą wspólny, kontekstowy system logiczny LMS/MCS. Autorzy tego systemu kierują się następującymi założeniami:

- Każdy z kontekstów zawiera fragment wiedzy o świecie, który da się scharakteryzować wg trzech wymiarów opisanych w rozdziale 4.1.1, czyli zakresem, poziomem szczegółowości i perspektywą.
- Kontekst nie jest rozumiany jako część struktury świata, lecz jako sposób ustrukturalizowania reprezentacji świata (kontekst jest *kognitywny*).
- Jak w przypadku PLC, dowodzenie odbywa się zawsze w jednym kontekście, jednak każdy z kontekstów może być wyrażony innym językiem, a nawet dysponować innymi regułami dowodzenia.
- Konteksty nie są po prostu niezależnymi reprezentacjami, ale różnymi obrazami tego samego świata. Poprawna logika dla dowodzenia kontekstowego musi więc dostarczać ogólnego sposobu opisanie związków pomiędzy procesami dowodzenia przebiegającymi w różnych kontekstach.
- Związki między kontekstami mogą być opisane tylko fragmentarycznie. Innymi słowy, nie jest możliwe przetransformowanie jednego kontekstu w drugi, gdyż każdy z nich jest oparty na nieznanym założeniu. Wynika z tego, że konteksty tworzą wielość reprezentacji i nie są redukowalne do jednej, uniwersalnej reprezentacji świata.

System spełnia też przytoczone w rozdziale 4.1.1 reguły wnioskowania kontekstowego: regułę lokalności i kompatybilności.

W systemie LMS/MCS konteksty nazywane są *podsystemami*. Niech $\{L_i \mid i \in I\}$ będzie rodziną języków dla podsystemów zdefiniowanych na zbiorze indeksów I . Każdy język L_i jest (formalnym) językiem używanym do opisu faktów w i -tym podsystemie. Niech \mathcal{M}_i będzie klasą wszystkich modeli języka L_i . Element $m \in \mathcal{M}_i$ nazywamy modelem lokalnym L_i .

Ponieważ podsystemy są wyrażone za pomocą odrębnych języków, nie istnieje przypadek, że ta sama formuła pojawia się w dwóch różnych podsystemach. Jednakże jest możliwe, że mogą pojawić się dwie formuły o tej samej składni. Jeśli nawet tak się stanie, należy mieć świadomość, że są one interpretowane w różnych lokalnych modelach, więc mają

różną semantykę. Aby odróżnić formułę φ w i -tym podsystemie od wystąpienia „tej samej” formuły φ w innych podsystemach, zastosowano oznaczenie $i:\varphi$. Formuła o postaci $i:\varphi$ nazywa się *formułą etykietowaną* i oznacza, że φ jest formułą zbudowaną w L_i . Dla dowolnego zbioru formuł etykietowanych S , $S_i = \{\varphi \mid i:\varphi \in S\}$.

Aby formalnie wyrazić opisane powyżej właściwości systemu kontekstowego, system LMS/MCS definiuje najpierw zasady interpretowania zdań zawartych w podsystemach tworzących system. Jest za to odpowiedzialna pierwsza część systemu, czyli LMS. Wprowadza ona do semantyki pojęcia *łańcucha kompatybilności* i *relacji kompatybilności*:

Definicja 4.4 (łańcuch kompatybilności)

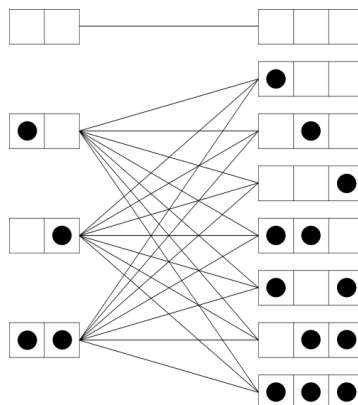
Łańcuch kompatybilności $\mu = \{M_i \subseteq M_i \mid i \in I \wedge \|M_i\| \leq 1\}$, czyli jest rodziną zbiorów modeli języka L_i takich, że każdy zbiór modeli M_i jest pusty lub jest jednoelementowy. M_i nazywany jest i -tym elementem łańcucha μ , co jest zapisywane $M_i = \mu(i)$. Łańcuch kompatybilności jest *niepusty*, gdy przynajmniej jeden z elementów jest niepusty. ■

Łańcuch kompatybilności gromadzi modele, co najwyżej po jednym dla każdego podsystemu. Jeśli dany łańcuch zawiera model dla podsystemu κ_i , to element M_i łańcucha jest zbiorem jednoelementowym. W przeciwnym przypadku M_i jest zbiorem pustym. Łańcuch reprezentuje więc zbiór „równoczesnych fotografii świata”, z których każda jest wykonana z punktu widzenia powiązanego podsystemu.

Definicja 4.5 (relacja kompatybilności i model LMS)

Relacja kompatybilności μ jest zbiorem łańcuchów kompatybilności. *Model LMS* jest relacją kompatybilności, która zawiera co najmniej jeden niepusty łańcuch kompatybilności. ■

Relacja kompatybilności formalizuje wszystkie dopuszczalne zbiory „równoczesnych fotografii świata”. Ponieważ podsystemy nie są względem siebie niezależne, niektóre kombinacje fotografii nie mogą się zdarzyć.



Rys. 4.6 Relacja kompatybilności dla magicznego pudełka (za [GS1994]).

W przykładzie 4.1 z magicznym pudełkiem z rozdziału 4.1.1 relacja kompatybilności dobrze oddaje zależność między wartościowaniami zdań w kontekstach obserwatorów Mr.1 i Mr.2. Jest ona przedstawiona na rys. 4.6. Można przyjąć, że kratki reprezentują zdania w języku kontekstu Mr.1 (czyli l dla sytuacji, gdy kulka jest z lewej strony i r dla sytuacji, gdy kulka jest z prawej strony) oraz w języku kontekstu Mr.2, czyli l , r i c (dla sytuacji, gdy kulka jest w środku). Gdy dane zdanie jest prawdziwe, w kratce z nią związanej znajduje się kropka i odwrotnie: gdy zdanie jest fałszywe, kratka jest pusta. Na rys. 4.6 widać, że istnieje

tylko jeden dopuszczalny łańcuch kompatybilności dla modelu przypisującego fałsz obu zdaniom z języka kontekstu Mr.1. Natomiast każde z wartościowań, przypisujących chociaż jednemu z tych zdań prawdę, jest kompatybilne aż z siedmioma sposobami wartościowań zdań z kontekstu Mr.2.

Definicja 4.6 (spełnianie i wynikanie)

Niech \models będzie relacją spełniania i wynikania znaną z klasycznej logiki zdaniowej. W modelu LMS definicja \models jest rozszerzana w następujący sposób:

- 1) $\forall \varphi \in L_i$ zbiór modeli M_i spełnia φ wtwg spełnia ją każdy model m z M_i .
Formalnie: $M_i \models \varphi \leftrightarrow \forall m \in M_i (m \models \varphi)$.
- 2) Łańcuch kompatybilności μ spełnia formułę etykietowaną $i:\varphi$ wtwg spełnia ją i -ty element tego łańcucha.
Formalnie: $\mu \models i:\varphi \leftrightarrow \mu(i) \models \varphi$.
- 3) Relacja kompatybilności μ spełnia formułę etykietowaną $i:\varphi$ wtwg spełnia ją każdy łańcuch kompatybilności μ z μ .
Formalnie: $\mu \models i:\varphi \leftrightarrow \forall \mu \in \mu (\mu \models i:\varphi)$.
- 4) W zbiorze modeli M_i ze zbioru formuł S_i wynika formuła φ wtwg φ wynika z S_i w każdym modelu m ze zbioru M_i .
Formalnie: $S_i \models_{M_i} \varphi \leftrightarrow \forall m \in M_i (m \models S_i \rightarrow m \models \varphi)$.
- 5) W łańcuchu kompatybilności μ ze zbioru formuł S wynika formuła etykietowana $i:\varphi$ wtwg $i:\varphi$ wynika z S_i w i -tym elemencie tego łańcucha, lub istnieje taki j -ty element, który nie spełnia S_j .
Formalnie: $S \models_{\mu} i:\varphi \leftrightarrow S_i \models_{\mu(i)} \varphi \vee \exists j \neq i (\mu(j) \not\models S_j)$.
- 6) W relacji kompatybilności μ ze zbioru formuł S wynika formuła etykietowana $i:\varphi$ wtwg $i:\varphi$ wynika z S w każdym łańcuchu kompatybilności μ z relacji μ .
Formalnie: $S \models_{\mu} i:\varphi \leftrightarrow \forall \mu \in \mu (S \models_{\mu} i:\varphi)$.
- 7) W klasie modeli LMS \mathcal{W} ze zbioru formuł S wynika formuła etykietowana $i:\varphi$, wtwg $i:\varphi$ wynika z S w każdej relacji kompatybilności μ z \mathcal{W} .
Formalnie: $S \models_{\mathcal{W}} i:\varphi \leftrightarrow \forall \mu \in \mathcal{W} (S \models_{\mu} i:\varphi)$.

■

Dla stwierdzeń na temat relacji \models autorzy zaadaptowali terminologię związaną ze spełnianiem. Tak więc mówi się, że μ spełnia φ w i lub, równoważnie, φ jest prawdziwe w $M_i = \mu(i)$, aby odzwierciedlić fakt, że $M_i \models \varphi$. Podobnie w przypadku relacji wynikania, mówi się na przykład, że z S wynika $i:\varphi$, aby odzwierciedlić fakt, że $S \models_{\mu} i:\varphi$. Analogiczna terminologia została zaadaptowana dla $S \models_{\mu} i:\varphi$ i $S \models_{\mathcal{W}} i:\varphi$.

Systemy wielokontekstowe MCS są klasą systemów dowodzenia dla LMS. Podstawowym pojęciem MCS jest *reguła pomostowa*.

Definicja 4.7 (reguła pomostowa)

Regułą pomostową (ang. *bridge rule*) na zbiorze indeksów I jest reguła dowodzenia o postaci:

$$\frac{\vdash i_1:\varphi_1 \dots i_n:\varphi_n}{\vdash i:\varphi} \quad (\text{BR})$$

gdzie $i_1, \dots, i_n, i \in I$, a $\varphi_1, \dots, \varphi_n, \varphi$ są formułami. Reguła pomostowa może być związana z *ograniczeniem* (*restriction*), opisującym warunki jej stosowalności. ■

Definicja 4.8 (system wielokontekstowy (MCS))

System wielokontekstowy (MCS) dla rodziny języków $\{L_i\}$, gdzie $i \in I$, jest parą $MS = (\{Th_i\}, R_{br})$, gdzie każde $Th_i = (L_i, Ak_i, R_i)$ jest teorią na języku L_i , o aksjomatach Ak_i i regułach dowodzenia R_i , a R_{br} jest zbiorem reguł pomostowych na I . ■

MCS jest uogólnieniem systemu dedukcji naturalnej. To uogólnienie polega na używaniu formuł oznaczanych indeksem języka, do którego należą. Mechanizm dedukcji systemu MCS łączy dwa typy reguł: lokalne reguły dowodzenia, należące do każdego zbioru R_i , i reguły pomostowe R_{br} . Reguły lokalne formalizują proces dowodzenia wewnątrz podsystemu (tzn. są stosowane jedynie do formuł oznaczonych tym samym indeksem), a reguły pomostowe formalizują przepływ wniosków między podsystemami.

Dowody w MSC są budowane, poczynając od skończonego zbioru założeń i aksjomatów, poprzez stosowanie skończonej liczby reguł lokalnych i pomostowych. Założenia i aksjomaty mogą należeć do różnych języków. Formuła $i:\varphi$ jest wyprowadzalna ze zbioru formuł S w systemie wielokontekstowym MS , symbolicznie $S \vdash_{MS} i:\varphi$, jeżeli istnieje dowód z końcową formułą $i:\varphi$ i początkowymi formułami należącymi do S . Formuła $i:\varphi$ jest twierdzeniem w MS , symbolicznie $\vdash_{MS} i:\varphi$, jeśli jest wyprowadzalna z pustego zbioru założeń.

Dowód w MSC może być postrzegany jako twór skomponowany z lokalnych dowodów częściowych w różnych językach, które są tworzone poprzez powtarzalne stosowanie reguł lokalnych, a każda para lub liczniejsza grupa dowodów częściowych jest łączona przez zastosowanie jednej lub większej liczby reguł pomostowych.

Przykład 4.3 (projekty medyczne w systemie MCS)

Rozpatrzmy ponownie przykład 4.2 ze str. 57. Przypomnijmy zawartość obu podsystemów (w poprzednim przykładzie nazywanych kontekstami). Podsystem *projekty*:

$$\text{projekty: } \forall x (\text{Projekt_dot_schorzeń_genetycznych}(x) \leftrightarrow (\text{Projekt}(x) \wedge \exists y (\text{dotyczy}(x, y) \wedge \text{Schorzenie_genetyczne}(y))))), \quad (4.5)$$

$$\text{projekty: } \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \leftrightarrow (\text{ProjektUE}(x) \wedge \exists y (\text{dotyczy}(x, y) \wedge \text{Mukowiscydoza}(y))))), \quad (4.6)$$

$$\text{projekty: } \forall x (\text{ProjektUE}(x) \rightarrow \text{Projekt}(x)), \quad (4.7)$$

$$\text{projekty: } \forall x (\exists y \text{ dotyczy}(x, y) \rightarrow \text{Projekt}(x)). \quad (4.8)$$

Podsystem *schorzenia*:

$$\text{schorzenia: } \forall x (\text{Zwłóknienie_genet}(x) \leftrightarrow (\text{Zwłóknienie}(x) \wedge \exists y (\text{spowodowane_przez}(x, y) \wedge \text{Przyczyny_genet}(y))))), \quad (4.10)$$

$$\text{schorzenia: } \forall x (\text{Mukowiscydoza}(x) \leftrightarrow (\text{Zwłóknienie_genet}(x) \wedge \exists y (\text{zlokalizowane_w}(x, y) \wedge \text{Trzustka}(y))))). \quad (4.11)$$

W przypadku systemu MCS nie ma potrzeby tworzyć dodatkowego podsystemu, gdyż zamiast aksjomatów przenoszących zostaną stworzone reguły pomostowe. Dlatego przyjmujemy, że podsystemem, w którym zacznie się i skończy dowodzenie, będzie podsystem *projekty*. Należy udowodnić formułę:

$$\text{projekty: } \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \rightarrow \text{Projekt_dot_schorzeń_genetycznych}(x)). \quad (4.31)$$

Komunikację między modułami będą zapewniały następujące reguły pomostowe:

$$\frac{\vdash \text{projekty: Mukowiscydoza}(x)}{\vdash \text{schorzenia: Mukowiscydoza}(x)} \quad (\text{PS1})$$

$$\frac{\vdash \text{schorzenia: } \exists y (\text{spowodowane_przez}(x, y) \wedge \text{Przyczyny_genet}(y))}{\vdash \text{projekty: Schorzenie_genetyczne}(x)} \quad (\text{PS2})$$

Dowód:

Dowód przebiega podobnie, jak w przykładzie 4.2 zamieszczonego w punkcie 4.1.3.5. Kroki od (4.14) do (4.17) są powtarzane. Następnie ze zdania:

$$\text{projekty: Mukowiscydoza}(y) \quad (4.17)$$

po zastosowaniu reguły PS1 otrzymujemy:

$$\text{schorzenia: Mukowiscydoza}(y). \quad (4.20)$$

Znów powtarzany jest krok (4.21) i (4.22), aby otrzymać zdanie:

$$\text{schorzenia: } \exists z (\text{spowodowane_przez}(y, z) \wedge \text{Przyczyny_genet}(z))). \quad (4.22)$$

Teraz można zastosować regułę pomostową SP1, po czym otrzymamy zdanie:

$$\text{projekty: Schorzenie_genetyczne}(y). \quad (4.25)$$

Powielając kroki (4.26) do (4.29), otrzymujemy:

$$\text{projekty: } \forall x (\text{ProjektUE_dot_mukowiscydozy}(x) \rightarrow \text{Projekt_dot_schorzeń_genetycznych}(x)). \quad (4.29)$$

Jest to właśnie zdanie, które należało udowodnić.

QED. ■

Powyższy przykład odbiega od poprzedniego tylko w tym, że zamiast wykorzystywać dodatkowy moduł i aksjomaty przenoszące używane są reguły pomostowe. Reguły te pozwalają na przenoszenie się z podsystemu do podsystemu. Pozostałe części dowodu prowadzone były lokalnie i nie różniły się niczym od analogicznych części z przykładu 4.2.

Autorzy systemu LMS/MCS rozwinęli go w stronę logiki pierwszego rzędu, tworząc *rozproszoną logikę pierwszego rzędu* (ang. *Distributed First Order Logics — DFOL*) [GS1998]. Semantyka dla logiki pierwszego rzędu jest inaczej zdefiniowana — zamiast funkcji wartościującej zdania mamy do czynienia z dziedziną rozumianą jako zbiór obiektów i przyporządkowaniem interpretacyjnym, które przypisuje nazwy ze słownika obiektom z dziedziny oraz relacjom między obiektami dziedziny. W związku z tym wymagania dla DFOL rozszerzają opisany wyżej zbiór wymagań głównie o punkty dotyczące semantyki:

- Podsystem posiada swoją dziedzinę.
- Podsystem posiada swoją lokalną interpretację.
- DFOL musi pozwolić na opisanie relacji między obiektami różnych dziedzin.

Spełnienie tych wymagań warunkuje możliwość dokonania poprawnego opisu semantyki systemu wielokontekstowego dla języków logiki pierwszego rzędu.

Niech znowu $\{L_i \mid i \in I\}$ będzie rodziną języków, a $\{k_i \mid i \in I\}$ zbiorem podsystemów zdefiniowanych na zbiorze indeksów I . Model m należący do klasy modeli \mathcal{M}_i języka L_i musi, zgodnie z wymaganiami, być parą (Δ, \bullet^J) , gdzie Δ_i jest lokalną dziedziną i -tego podsystemu, a \bullet^J jest przyporządkowaniem interpretacyjnym dla języka. Ponieważ każdy podsystem jedynie częściowo opisuje wiedzę o dziedzinie, więc z każdym językiem L_i jest powiązany

zbiór *możliwych modeli* (*possible models*) $M_i \subseteq \mathcal{M}_i$. Drugim ważnym elementem logiki DFOL jest *relacja dziedzinowa* r_{ij} , która jest definiowana jako podzbiór iloczynu kartezjańskiego $\Delta_i \times \Delta_j$. Relacja ta odzwierciedla, w jaki sposób obiekty z dziedziny Δ_i są reprezentowane w dziedzinie Δ_j . Jest to relacja, a nie funkcja, ponieważ mogą zaistnieć sytuacje, gdy jeden obiekt w Δ_i będzie odpowiadał wielu obiektom w Δ_j i odwrotnie — wiele obiektów w Δ_i może mieć jednego przedstawiciela w Δ_j . Opierając się na pojęciach możliwych modeli i relacji dziedzinowej, autorzy definiują model dla DFOL.

Definicja 4.9 (model (DFOL))

Modelem \mathcal{M} dla rodziny języków $\{L_i\}$ jest para $(\{M_i\}, \{r_{ij}\})$, gdzie $M_i \subseteq \mathcal{M}_i$ jest zbiorem możliwych modeli L_i z tą samą dziedziną Δ_i , a r_{ij} jest relacją dziedzinową z i do j . ■

System DFOL stał się teoretyczną podstawą dla grupy metod tworzenia ontologii kontekstowych oznaczanych skrótem DDL. Zostaną one opisane w dalszej części niniejszej rozprawy.

4.2 Modularyzacja ontologii w Semantic Web

Inicjatywa Semantic Web spowodowała, że w Internecie pojawiło się wiele różnych ontologii, które opisywały wiele różnych dziedzin. Z punktu widzenia informacji w nich zawartej jest to wielka skarbnica wiedzy. Jednakże z technicznego punktu widzenia informacja ta ma ograniczoną wartość, gdyż ontologie te jedynie w niewielkim stopniu nadają się do powtórnego użycia. Potrzeba znalezienia metod, które umożliwiłyby wykorzystanie tej skarbnicy wiedzy, stała się silnym bodźcem do opracowania konkretnych metod wykorzystujących w praktyce opisane w poprzednich podrozdziałach rozwiązania teoretyczne. Problemem zajęło się wielu badaczy z różnych krajów i dzięki ich wysiłkowi powstało wiele propozycji modularyzacji. Większość z nich jest nakierowana na rozwiązanie któregoś z wymienionych we wstępie do niniejszej pracy problemów referencyjnych. Niektóre są dość mocno wyspecjalizowane. Do takich technik należą te opisane w [DSM2006], [SK2004], [SR2006]. Zostały one zaproponowane, by ułatwić powtórne wykorzystanie istniejących dużych ontologii i poprawić wydajność wnioskowania. Polegają na automatycznym wydzieleniu odpowiednich fragmentów poprzez analizę leksykalną zapytania. Problemem jest oczywiście określenie, jaki fragment ontologii powinien zostać wydzielony. W tym celu stosowane są różne metody. W [SK2004] i [SR2006] ontologię przedstawia się jako graf modelujący hierarchię conceptów i powiązania między conceptami wyznaczone przez dziedziny i zakresy ról. W grafie tym poszukuje się węzłów związanych z nazwami użytymi w zapytaniu. Następnie dokonuje się ekstrakcji tych węzłów wraz z ich znaczącym otoczeniem. Otoczenie to wyznacza się albo poprzez zastosowanie metod statystycznych pozwalających wyliczyć wagi połączeń między węzłami (w [SK2004]), albo poprzez trawersowanie ścieżek algorytmem parametryzowanym przez użytkownika (w [SR2006]). Przykładowym parametrem jest odległość danego węzła od węzła reprezentującego nazwę użytą w zapytaniu. Takie techniki ekstrakcji fragmentu ontologii nie mogą być dokładne, ale są wydajne. Z kolei metoda zaproponowana w [DSM2006] polega na zastosowaniu algorytmu wnioskującego do analizy semantycznej pozwalającej wyodrębnić żądany fragment. Jest to metoda dokładniejsza, lecz charakteryzuje się wyższą złożonością.

Inną metodę zaprezentowano w [R2003]. Jej celem jest umożliwienie podziału ontologii w celu usprawnienia procesu jej tworzenia. Ponieważ częścią każdej ontologii jest jakaś taksonomia, autor na niej właśnie opiera modularyzację: modułami mogą być jedynie gałęzie drzewa taksonomii. Zestaw reguł zaproponowanych w [R2003] wspiera projektantów dokonujących podziału. W efekcie oprócz rozłącznych modułów otrzymuje się dodatkowo zbiór aksjomatów opisujących wszelkie powiązania między modułami. Aksjomaty te

nazwane zostały *definicjami łączącymi* (ang. *linking definitions*) i *ograniczeniami łączącymi* (ang. *linking restrictions*). Taki podział obowiązuje jedynie w czasie tworzenia ontologii. Po tym okresie moduły są łączone, dodawane są definicje i ograniczenia łączące i efektem jest duża, monolityczna ontologia.

Przytoczone tu przykłady mają na celu zilustrować, jak różnymi drogami autorzy technik modularyzacji pragną uzyskać określony cel. Mają one wspólny mianownik: nie zmieniają algorytmów wnioskowania. Przekształceniu podlega sama ontologia i ma to miejsce jedynie na etapie przygotowawczym. Po przekształceniu działanie polega na zwykłym wnioskowaniu z jednej ontologii. Techniki te mają jeszcze jedną wspólną cechę: są one pragmatycznie zorientowane na rozwiązanie któregoś z problemów referencyjnych i nie mają podstawy teoretycznej.

W dalszej części rozdziału zaprezentowane zostaną dwie techniki, które nie mają tych wad. Opierają się one na opisanym w punkcie 4.1.4 systemie LMS/MCS. Tak więc bazują na dobrze ugruntowanym formalizmie opisującym rolę kontekstów w logice. Ich celem głównym nie jest więc rozwiązanie jakiegoś konkretnego problemu, lecz wprowadzenie kontekstualizacji baz wiedzy rozumianej jako cecha strukturalna. Oczywiście autorzy tych technik kierują się przekonaniem, że przy okazji zostanie też rozwiązana większość z problemów referencyjnych. W punkcie 4.2.1 przedstawiono rozproszoną logikę opisową, a w punkcie 4.2.2 formalizm o nazwie \mathcal{E} -Connections. Są to techniki pokrewne, jednak tę samą ideę realizują stosując zupełnie inne rozwiązania techniczne.

4.2.1 Rozproszona logika opisowa (DDL)

Rozproszona logika opisowa (ang. *Distributed Description Logics* — *DDL*) została zaproponowana przez A. Borgidę i L. Serafiniego ([BS2002], [BS2003]) jako projekcja systemu DFOL na logikę opisową. Mówiąc w uproszczeniu, system ten, bazując na LMS/MCS, wprowadza pojęcie rozproszonej interpretacji polegającej na tym, że wiedza zawarta w każdym z modułów (podsystemów) posiada własny zbiór modeli, jednakże zbiory te ograniczane są regułami pomostowymi (patrz punkt 4.1.4) do zbiorów kompatybilnych względem siebie. Ponieważ każdy z modułów może być traktowany jako osobna ontologia, która może być wykorzystywana niezależnie, kluczowym elementem integrującym są reguły pomostowe. Zasady te zostały zachowane w DDL, jednak wprowadzono kilka zmian. Pierwsza polega na tym, że dopuszczalnymi językami użytymi do opisu wiedzy w modułach są języki stworzone na bazie logik opisowych. Druga, uwzględniająca fakt, że baza wiedzy w logice opisowej składa się z terminologii i zbioru faktów, polega na podziale reguł pomostowych na dwie grupy: grupę reguł opisujących zależności między conceptami (ta grupa zachowała nazwę reguł pomostowych) oraz grupę opisującą zależności wiążące osobniki (*reguły odpowiedniości osobników*).

Tak jak w DFOL, ponieważ każdy z modeli może mieć inną dziedzinę, syntaktyczne oddziaływanie reguł musi być wsparte opisem zależności między dziedzinami łączonych modułów. Symbolicznie relacja $r_{ij} \subseteq \Delta_i \times \Delta_j$ służy jako opis tej zależności.

Dla danego niepustego zbioru indeksów I oraz zbioru języków logik opisowych $\{\mathcal{L}_i \mid i \in I\}$ powyższe reguły definiuje się w następujący sposób:

Definicja 4.10 (reguły pomostowe)

Dla danej nazwy conceptu C pochodzącej ze słownika \mathcal{L}_i i nazwy conceptu D ze słownika \mathcal{L}_j regułami pomostowymi z i do j są wyrażenia:

$$i:C \xrightarrow{\sqsubseteq} j:D \quad \text{reguła wstawiająca (into rule)}$$

$$i:C \xrightarrow{\sqsubseteq} j:D \quad \text{reguła wchłaniająca (onto rule)}$$

Reguła wstawiająca deklaruje, że osobniki będące wystąpieniami konceptu C z i -tego modułu odpowiadają jedynie osobnikom będącym wystąpieniami konceptu D w j -tym module. Z kolei reguła wchłaniająca działa odwrotnie: osobniki, które są wystąpieniami konceptu D w j mogą odpowiadać jedynie osobnikom będącym wystąpieniami konceptu C w module i .

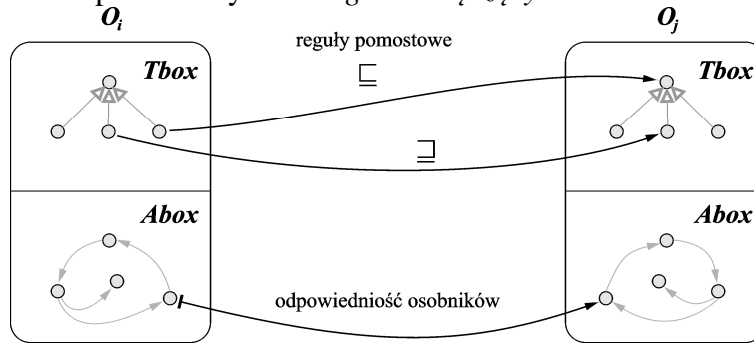
Reguły odpowiedności osobników definiuje się tak:

Definicja 4.11 (reguły odpowiedności osobników)

Dla danej nazwy osobnika a pochodzącej ze słownika \mathcal{L}_i oraz b ze słownika \mathcal{L}_j regułą odpowiedności osobników (ang. *individual correspondence*) z i do j nazywamy wyrażenie:

$$i:a \longmapsto j:b$$

Jeśli nie będzie istniała potrzeba rozdzielania reguł pomostowych od reguł odpowiedności osobników, będą one wspólnie nazywane *regułami łączącymi*.



Rys. 4.7 Przykład ontologii modularnej wykonanej za pomocą techniki DDL. O_i jest modulem źródłowym, a O_j modulem docelowym (na podst. [ST2007]).

Rozproszoną bazę wiedzy opartą na logice opisowej, czyli system DDL, definiuje się uwzględniając istnienie modułów i dodając zbiory reguł pomostowych oraz reguł odpowiedności osobników.

Definicja 4.12 (rozproszona baza wiedzy (DDL))

Rozproszoną bazę wiedzy nazywamy parę $(\mathcal{T}, \mathcal{A})$, gdzie:

- 1) rozproszony Tbox \mathcal{T} jest parą (\mathbf{T}, \mathbf{R}) , na którą składa się zbiór Tboxów modułów $\mathbf{T} = \{T_i \mid i \in I\}$ oraz zbiór reguł pomostowych $\mathbf{R} = \{R_{ij} \mid i \neq j \in I\}$; dla każdego $k \in I$ Tbox T_k musi być wyrażony w \mathcal{L}_k ; dla każdej reguły pomostowej postaci $i:A \xrightarrow{\sqsubseteq} j:B$ oraz $i:A \xrightarrow{\sqsupseteq} j:B$ w R_{ij} koncept A musi znajdować się w słowniku \mathcal{L}_i , a koncept B w słowniku \mathcal{L}_j ,
- 2) rozproszony Abox \mathcal{A} jest parą (\mathbf{A}, \mathbf{C}) , na którą składa się zbiór Aboxów modułów $\mathbf{A} = \{A_i \mid i \in I\}$ oraz zbiór reguł odpowiedności osobników $\mathbf{C} = \{C_{ij} \mid i \neq j \in I\}$; dla każdego $k \in I$ Abox A_k musi być wyrażony w \mathcal{L}_k ; dla każdej reguły odpowiedności osobników postaci $i:a \longmapsto j:b$ w C_{ij} nazwa osobnika a musi znajdować się w słowniku \mathcal{L}_i , a nazwa osobnika b w słowniku \mathcal{L}_j .

Moduły są w takim ujęciu ontologiami wyrażonymi za pomocą różnych języków z rodziny języków logiki opisowej. Mogą być używane niezależnie w ten sposób, że dany algorytm

wnioskujący nie będzie brał pod uwagę istnienia zbiorów \mathbf{R} i \mathbf{C} . Jednak chcąc traktować te moduły jako części systemu, algorytm wnioskujący musi uwzględnić istnie reguł łączących. Ich istnienie ma na celu ograniczenie branych pod uwagę modeli do tych tylko, które dopuszcza semantyka reguł. Rysunek 4.7 ilustruje sposób powiązania modułów w rozproszonej bazie wiedzy tworzącej system DDL.

Takie ujęcie jest bardzo wygodne w Semantic Web. W internecie istnieje wiele ontologii traktowanych jako niezależne bazy wiedzy. Można wybrać dowolne z nich i powiązać dowolnym zbiorem reguł łączących w system DDL. Co więcej, różne zbiory reguł działające na tych samych lub zazębiających się zbiorach ontologii mogą wyznaczać różne systemy DDL. Do tych systemów można dołączać nowe ontologie własnego autorstwa, które doprecyzowują opis danej dziedziny z punktu widzenia wygodnego dla danej aplikacji.

Semantyka systemu DDL uwzględnia zasady lokalności i kompatybilności wprowadzone przez autorów systemu LMS/MCS. Interpretacja systemu rozproszonego jest więc przede wszystkim tworzona przez interpretacje lokalne modułów. Semantyka reguł łączących ogranicza następnie zbiór tych interpretacji w ten sposób, aby nie kolidowały one z wiedzą z modułu źródłowego widoczną z poziomu modułu docelowego.

Definicja 4.13 (rozproszona interpretacja)

Rozproszoną interpretacją (ang. *distributed interpretation*) rozproszonej bazy wiedzy nazywamy parę $(\{\mathcal{I}_i \mid i \in I\}, r)$. Pierwszy element pary zawiera interpretacje \mathcal{I}_i dla \mathcal{L}_i na dziedzinie $\Delta^{\mathcal{I}_i}$. Drugi element pary jest funkcją przyporządkowującą każdej parze $i \neq j \in I$ relację dziedzinową $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$. ■

Z definicji tej wynika, że aby uzyskać jedną interpretację rozproszoną, należy wybrać po jednej interpretacji lokalnej dla każdego modułu oraz określić dla każdej pary modułów co najwyżej jedną relację dziedzinową. Takie ujęcie semantyki wywodzi się z idei łańcuchów kompatybilności opisanych w rozdziale (4.1.4). Łańcuch taki da się utworzyć tylko z tak dobranych interpretacji, by wzajemnie sobie nie zaprzeczały. Następne definicje doprecyzowują właściwości tych interpretacji. Dla uproszczenia w dalszym ciągu rozważań przyjmuje się, że zapis $r_{ij}(d)$, gdzie $d \in \Delta^{\mathcal{I}_i}$, oznacza zbiór $\{d' \in \Delta^{\mathcal{I}_j} \mid (d, d') \in r_{ij}\}$, a zapis $r_{ij}(D)$, gdzie $D \subseteq \Delta^{\mathcal{I}_i}$, oznacza zbiór $\cup_{d \in D} r_{ij}(d)$.

Definicja 4.14 (d-spełnialność)

D-spełnialność (ang. *d-satisfiability*) \models_d jest relacją między rozproszoną interpretacją $\mathcal{J} = (\{\mathcal{I}_i \mid i \in I\}, \{r_{ij} \mid i \neq j \in I\})$ i jej składnikami a rozproszoną bazą wiedzy $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ i jej składnikami. Interpretacja \mathcal{J} spełnia bazę \mathcal{K} , co jest zapisywane $\mathcal{J} \models_d \mathcal{K}$, jeśli:

- 1) spełnia rozproszony Tbox $\mathcal{T} = (\mathbf{T}, \mathbf{R})$, co jest zapisywane $\mathcal{J} \models_d \mathcal{T}$. Oznacza to, że:
 - a) spełnia zbiór lokalnych Tboxów $\mathbf{T} = \{T_i \mid i \in I\}$, co jest zapisywane $\mathcal{J} \models_d \mathbf{T}$;
 - i) $\mathcal{J} \models_d T_i$, jeśli spełnia każdy $T_i \in \mathbf{T}$, co jest zapisywane $\mathcal{J} \models_d T_i$;
 - ii) $\mathcal{J} \models_d T_i$, jeśli spełnia każdy aksjomat $\varphi \in T_i$, co jest zapisywane $\mathcal{J} \models_d i:\varphi$;
 - iii) $\mathcal{J} \models_d i:\varphi$, jeśli φ jest standardowo spełniane przez odpowiednią interpretację lokalną, co jest zapisywane $\mathcal{I}_i \models \varphi$;
 - b) spełnia zbiór reguł pomostowych $\mathbf{R} = \{R_{ij} \mid i \neq j \in I\}$, co jest zapisywane $\mathcal{J} \models_d \mathbf{R}$;
 - i) $\mathcal{J} \models_d R_{ij}$, jeśli spełnia każdy zbiór reguł pomostowych $R_{ij} \in \mathbf{R}$, co jest zapisywane $\mathcal{J} \models_d R_{ij}$;
 - ii) $\mathcal{J} \models_d R_{ij}$, jeśli spełnia każdą regułę wstawiającą, co jest zapisywane $\mathcal{J} \models_d i:A \sqsubseteq j:G$, i wchłaniającą, co jest zapisywane $\mathcal{J} \models_d i:B \sqsupseteq j:H$, należąca do R_{ij} ;

- iii) $\mathcal{J} \models_d i:A \xrightarrow{\sqsubseteq} j:G$, jeśli $r_{ij}(A^{T_i}) \subseteq G^{T_j}$;
 - iv) $\mathcal{J} \models_d i:B \xrightarrow{\supseteq} j:H$, jeśli $r_{ij}(B^{T_i}) \supseteq H^{T_j}$;
- 2) spełnia rozproszony Abox $\mathcal{A} = (\mathbf{A}, \mathbf{C})$ co jest zapisywane $\mathcal{J} \models_d \mathcal{A}$. Oznacza to, że:
- a) spełnia zbiór lokalnych Aboxów $\mathbf{A} = \{A_i \mid i \in I\}$, co jest zapisywane $\mathcal{J} \models_d \mathbf{A}$;
 - i) $\mathcal{J} \models_d \mathbf{A}$, jeśli spełnia każdy $A_i \in \mathbf{A}$, co jest zapisywane $\mathcal{J} \models_d A_i$;
 - ii) $\mathcal{J} \models_d A_i$, jeśli spełnia każdą asercję $\varphi \in A_i$, co jest zapisywane $\mathcal{J} \models_d i:\varphi$;
 - iii) $\mathcal{J} \models_d i:\varphi$, jeśli φ jest standardowo spełniane przez odpowiednią interpretację lokalną, co jest zapisywane $\mathcal{I}_i \models \varphi$;
 - b) spełnia zbiór reguł odpowiedniości osobników $\mathbf{C} = \{C_{ij} \mid i \neq j \in I\}$, co jest zapisywane $\mathcal{J} \models_d \mathbf{C}$;
 - i) $\mathcal{J} \models_d \mathbf{C}$, jeśli spełnia każdy zbiór reguł odpowiedniości osobników $C_{ij} \in \mathbf{C}$, co jest zapisywane $\mathcal{J} \models_d C_{ij}$;
 - ii) $\mathcal{J} \models_d C_{ij}$, jeśli spełnia każdą regułę odpowiedniości osobników należącą do C_{ij} , co jest zapisywane $\mathcal{J} \models_d i:a \longmapsto j:b$;
 - iii) $\mathcal{J} \models_d i:a \longmapsto j:b$, jeśli $b^{T_j} \in r_{ij}(a^{T_i})$.

■

Jeśli dana rozproszona interpretacja d-spełnia daną rozproszoną bazę wiedzy, to mówi się, że jest jej modelem. Z definicji widać wyraźnie, w jaki sposób lokalna interpretacja \mathcal{I}_i wpływa poprzez zbiór reguł łączących na zbiór możliwych interpretacji modułu j , które mogą zostać zaakceptowane jako \mathcal{I}_j . Jeśli w danej rozproszonej bazie wiedzy zdefiniowano reguły pomostowe

$$\begin{array}{ccc} i:A & \xrightarrow{\sqsubseteq} & j:G \\ i:B & \xrightarrow{\supseteq} & j:H \end{array}$$

oraz w T_i istnieje aksjomat $B \sqsubseteq A$, to jako \mathcal{I}_j może być zaakceptowana tylko taka interpretacja lokalna modułu j , w której wszystkie osobniki będące wystąpieniami H są również wystąpieniami G . Jest to więc równoznaczne z istnieniem w T_i aksjomatu $H \sqsubseteq G$. Niech ilustracją tego będzie następujący przykład.

Przykład 4.4 (projekty medyczne w systemie DDL)

Rozpatrzmy ponownie moduły z przykładu 4.2 (str. 57) i 4.3 (str. 62). Treść tych modułów nie ulega zmianie, jednakże aksjomaty zmieniają swoją formę, gdyż będą zapisane przy użyciu syntaktyki logiki opisowej. Moduł *projekty* (oznaczony tu symbolem 1) zawiera:

$$1:(Projekt_dot_schorzeń_genetycznych \equiv Projekt \sqcap \exists dotyczy.Schorzenia_genetyczne), \quad (4.32)$$

$$1:(ProjektUE_dot_mukowiscydozy \equiv ProjektUE \sqcap \exists dotyczy.Mukowiscydoza), \quad (4.33)$$

$$1:(ProjektUE \sqsubseteq Projekt), \quad (4.34)$$

$$1:(\exists dotyczy.T \sqsubseteq Projekt). \quad (4.35)$$

Moduł *schorzenia* (2) zawiera następujące aksjomaty:

$$2:(Zwłóknienie_genet \equiv Zwłóknienie \sqcap \exists spowodowane_przez.Przyczyny_genet), \quad (4.36)$$

$$2:(Mukowiscydoza \sqsubseteq Zwłóknienie_genet \sqcap \exists zlokalizowane_w.Trzustka). \quad (4.37)$$

Formuła, którą należy udowodnić, zapisana w logice opisowej, brzmi:

$$1:(ProjektUE_dot_mukowiscydozy \sqsubseteq Projekt_dot_schorzeń_genetycznych). \quad (4.38)$$

Reguły pomostowe, których zdefiniowanie jest konieczne do udowodnienia powyższej formuły, są następujące:

$$R_{12} = \{$$

$$1:Mukowiscydoza \xrightarrow{\sqsubseteq} 2:Mukowiscydoza \quad (BR1)$$

$$1:Schorzenie_genetyczne \xrightarrow{\sqsupseteq} 2:\exists spowodowane_przez.Przyczyny_genet \quad (BR2)$$

$$\}$$

Reguły pomostowe w DDL różnią się od tych z systemu MCS. Przede wszystkim nie wyrażają zależności między zdaniami, lecz między konceptami. Zachowują więc główną zasadę logiki opisowej, czyli deskrypcyjność. Pierwsza z nich jest regułą wstawiającą, druga jest regułą wchłaniającą. Tak sformułowane wystarczą, aby algorytm *tableau* stwierdził, że w module *projekty* koncept *Mukowiscydoza* jest subsumowany przez koncept *Schorzenie_genetyczne*. To z kolei powoduje, że *Projekt_dot_schorzeń_genetycznych* musi subsumować *ProjektUE_dot_mukowiscydozy*. ■

System DDL jest wciąż rozwijany. Rozwój ten koncentruje się na ułatwieniu powtórnego użycia istniejących w Internecie ontologicznych baz wiedzy. Podstawowym problemem jest w takim wypadku heterogeniczność ujęcia zjawisk w ramach opisywanej dziedziny, przejawiająca się w tym, że istnieją encje, które jedna ontologia opisuje konceptem, a druga rolą. Aby objąć systemem takie przypadki, autorzy proponują dołączenie dodatkowych typów relacji dziedzinowych oraz reguł łączących. *Relacja dziedzinowa koncept-rola* cr_{ij} łączy w pary obiekty z dziedziny Δ_i z *dopuszczalnymi trójkami* (ang. *admissible triples*) z dziedziny Δ_j . Dopuszczalne trójki, to trójki obiekt-rola-obiekt, zapisywane (x_1, R, x_2) , gdzie $x_1, x_2 \in \Delta_j$, a R jest nazwą roli ze słownika języka \mathcal{L}_j . Z kolei *relacja dziedzinowa rola-koncept* łączy w pary dopuszczalne trójki z dziedziny Δ_i z obiektami z dziedziny Δ_j . Odpowiednio do tych relacji definiuje się *reguły pomostowe koncept-rola*, jako wyrażenie postaci $i:C \xrightarrow{\sqsubseteq} j:R$ (lub $i:C \xrightarrow{\sqsupseteq} j:R$), oraz *reguły pomostowe rola-koncept*, jako wyrażenie postaci $i:R \xrightarrow{\sqsubseteq} j:C$ (lub $i:R \xrightarrow{\sqsupseteq} j:C$).

4.2.2 \mathcal{E} -Connections

Technika *\mathcal{E} -connections* (\mathcal{E} — epsilon) została zaproponowana ([KWZ2002], [KLWZ2003]) jako technika łączenia ontologii wyrażonych w ramach systemów sklasyfikowanych jako *Abstrakcyjne Systemy Opisowe* (ang. *Abstract Description Systems* — ADS). Charakteryzują się one tym, że ich języki dają się przełożyć na *Abstrakcyjny Język Opisowy* (ang. *Abstract Description Language* — ADL), a modele na *Abstrakcyjny Model Opisowy* (ang. *Abstract Description Model* — ADM). W [KLWZ2004] zostało pokazane, w jaki sposób dokonać takiej translacji w przypadku logiki opisowej. W kolejnych pracach (np. w [GPS2004]) pojawiły się propozycje rozszerzenia formalizmów logiki opisowej w ten sposób, aby można było za ich pomocą łączyć ontologie w *połączoną bazę wiedzy* (ang. *combined knowledge base*). Prace te spowodowały, że technika \mathcal{E} -connections stała się użyteczna w ramach inicjatywy Semantic Web. Na przykład znany edytor ontologii SWOOP (<http://www.mindswap.org/2004/SWOOP/>) został wyposażony w możliwość definiowania połączonych baz wiedzy (<http://www.mindswap.org/2004/multipleOnt/>).

Ogólnie mówiąc, technika ta polega na ograniczaniu zbioru możliwych interpretacji modułu docelowego poprzez wiązanie go z modułem źródłowym za pomocą *relacji łączących* (ang. *link relations*). Relacje te nie są, jak w przypadku systemu DDL, regułami poszerzającymi zbiór reguł sterujących systemem dowodzenia. Są one zdaniem wymagającymi interpretacji.

Niech, jak poprzednio, I będzie zbiorem indeksów, na którym zdefiniowany jest system modularny. Aby ograniczyć zbiór możliwych interpretacji i -tego modułu, wykorzystując zależności zdefiniowane w ramach j -tego modułu, tworzy się zbiór ij -relacji oznaczany symbolem \mathcal{E}_{ij} . W formalizmach o wyższej ekspresywności (zawierających w swoim oznaczeniu literę \mathcal{H}) istnieje możliwość stworzenia *połączonego Lboxa* (ang. *combined Lbox*), który zawiera aksjomaty opisujące relację subsumcji pomiędzy relacjami łączącymi.

Definicja 4.15 (połączony Lbox)

Niech $\{\mathcal{E}_{ij} \mid i \neq j \in I\}$ będzie zbiorem rozłącznych zbiorów relacji łączących. L_{ij} jest skończonym zbiorem aksjomatów opisujących relację subsumcji ij -relacji. Dla ij -relacji p i q w formalizmach logiki opisowej o ekspresji \mathcal{H} dopuszcza się aksjomaty postaci $p \sqsubseteq q$. W językach o niższej ekspresywności zbiory L_{ij} są puste. Połączonym Lboxem jest skończony zbiór $\mathbf{L} = \{L_{ij} \mid i \neq j \in I\}$. ■

W systemie modularnym dla danego i -tego modułu, koncepty są nazywane *i-konceptami*, role *i-rolami*, a osobniki *i-osobnikami*. Obecność relacji łączących wpływa na język i -tego modułu w ten sposób, że dozwolone jest tworzenie definicji i -konceptów złożonych używających nazw ij -relacji łączących oraz j -konceptów, j -ról i j -osobników. Tak więc definicja 2.5 z rozdziału 2.2.2 opisująca sposób definiowania konceptów jest rozszerzona o człon:

Definicja 4.16 (definicje konceptów)

Jeśli G jest j -konceptem, a p jest ij -relacją, to wyrażenia postaci $\exists p.G, \forall p.G, \leq n p.G$ i $\geq n p.G$ są i -konceptami. ■

Jako pełnoprawne koncepty te mogą być użyte zarówno w aksjomatach tworzonych w Tboxie, jak i asercjach tworzonych w Aboxie i -tego modułu. Abox może też być rozszerzony o asercje wyrażające istnienie wystąpienia relacji łączącej między osobnikiem i -tego i j -tego modułu.

Definicja 4.17 (dozwolone asercje)

Jeśli a jest i -osobnikiem, x jest j -osobnikiem, p jest ij -relacją, a q jest ji -relacją, to wyrażenia postaci $p(a, x)$ i $q(x, a)$ są dozwolonymi asercjami w Aboxie i -tego modułu. ■

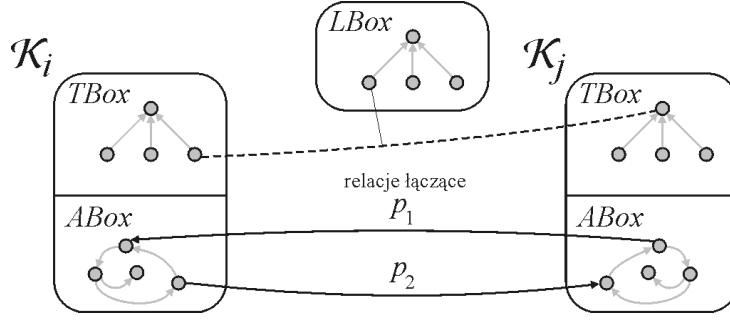
Tak więc można powiedzieć, że tworzy się nowy „superjęzyk”, który jest sumą języków poszczególnych modułów $\bigcup_{i \in I} \mathcal{L}_i$ rozszerzoną o dodatkowe elementy.

Połączona baza wiedzy jest złożeniem *połączonego Tboxa* (ang. *combined Tbox*), *połączonego Aboxa* (ang. *combined Abox*) i połączonego Lboxa.

Definicja 4.18 (połączona baza wiedzy)

Połączona baza wiedzy jest trójką $\mathcal{K} = (\mathbf{T}, \mathbf{A}, \mathbf{L})$, gdzie $\mathbf{T} = \{T_i \mid i \in I\}$ jest *połączonym Tboxem*, $\mathbf{A} = \{A_i \mid i \in I\}$ jest *połączonym Aboxem*, a $\mathbf{L} = \{L_{ij} \mid i \neq j \in I\}$ jest *połączonym Lboxem*. ■

Na rys. 4.8 zilustrowany jest sposób łączenia modułów w połączonej bazie wiedzy.



Rys. 4.8 Przykład modularnej kontekstowej wykonanej za pomocą techniki \mathcal{E} -Connections. LBox zawiera opis relacji łączących z \mathcal{E}_{ij} , a p_1 i p_2 symbolizują wystąpienia relacji łączących.

Semantyka systemu bazuje na pojęciu *połączonej interpretacji* (ang. *combined interpretation*).

Definicja 4.19 (połączona interpretacja)

Połączona interpretacja jest strukturą $\mathcal{J} = (\{\Delta^{I_i} \mid i \in I\}, \{\bullet^{I_i} \mid i \in I\}, \{\bullet^{I_{ij}} \mid i \neq j \in I\})$, gdzie Δ^{I_i} jest dziedziną interpretacyjną i -tego modułu taką, że dla $i \neq j$ obowiązuje równość $\Delta^{I_i} \cap \Delta^{I_j} = \emptyset$ i gdzie \bullet^{I_i} , $\bullet^{I_{ij}}$ są przyporządkowaniami interpretacyjnymi działającymi w sposób następujący:

- każdemu i -konceptowi atomowemu A przyporządkowują zbiór $A^{I_i} \subseteq \Delta^{I_i}$,
- każdej i -roli atomowej R przyporządkowują zbiór par $R^{I_i} \subseteq \Delta^{I_i} \times \Delta^{I_i}$,
- każdemu i -osobnikowi a przyporządkowują element $a^{I_i} \in \Delta^{I_i}$,
- każdej ij -relacji p przyporządkowują zbiór par $p^{I_{ij}} \subseteq \Delta^{I_i} \times \Delta^{I_j}$,

Ponadto koncepty złożone są interpretowane przez funkcje \bullet^{I_i} i $\bullet^{I_{ij}}$ w następujący sposób. W zakresie i -konceptów nie wykorzystujących ról łączących interpretacja jest zgodna z opisaną w definicji 2.9. W zakresie konceptów wykorzystujących role łączące obowiązują równości:

- $(\exists p.Z)^{I_i} = \{a \in \Delta^{I_i} \mid \exists x \in \Delta^{I_j} ((a, x) \in p^{I_{ij}} \wedge x \in Z^{I_j})\}$
- $(\forall p.Z)^{I_i} = \{a \in \Delta^{I_i} \mid \forall x \in \Delta^{I_j} ((a, x) \in p^{I_{ij}} \rightarrow x \in Z^{I_j})\}$
- $(\geq np.Z)^{I_i} = \{a \in \Delta^{I_i} \mid \|\{x \in \Delta^{I_j} \mid (a, x) \in p^{I_{ij}} \wedge x \in Z^{I_j}\}\| \geq n\}$
- $(\leq np.Z)^{I_i} = \{a \in \Delta^{I_i} \mid \|\{x \in \Delta^{I_j} \mid (a, x) \in p^{I_{ij}} \wedge x \in Z^{I_j}\}\| \leq n\}$

Gdzie p jest ij -relacją, a Z jest j -konceptem. ■

W tym przypadku mamy do czynienia z jedną interpretacją, a nie ze zbiorem interpretacji lokalnych, jak to było w przypadku systemu DDL. Interpretacja ta łączy dziedziny i przyporządkowania interpretacyjne związane z poszczególnymi modułami lub ich parami. Nie występuje też relacja dziedziniowa, gdyż od początku zakłada się rozłączność dziedzin. Dzięki temu uproszczeniu także spełnianie w połączonej bazie wiedzy jest zdefiniowane nieco prościej niż w systemie DDL.

Definicja 4.20 (spełnianie w połączonej bazie wiedzy)

Połączona interpretacja $\mathcal{J} = (\{\Delta^{I_i} \mid i \in I\}, \{\bullet^{I_i} \mid i \in I\}, \{\bullet^{I_{ij}} \mid i \neq j \in I\})$ spełnia połączoną bazę wiedzy $\mathcal{K} = (\mathbf{T}, \mathbf{A}, \mathbf{L})$, co jest zapisywane $\mathcal{J} \models \mathcal{K}$, jeśli:

- 1) spełnia połączony Tbox $\mathbf{T} = \{T_i \mid i \in I\}$, co jest zapisywane $\mathcal{J} \models \mathbf{T}$ i oznacza, że spełnia każdy $T_i \in \mathbf{T}$:
 - a) \mathcal{J} spełnia $T_i \in \mathbf{T}$, co jest zapisywane $\mathcal{J} \models T_i$ wtwg spełnia każdy aksjomat postaci $X_i \sqsubseteq Y_i$ z T_i , gdzie X_i, Y_i są albo i -konceptami, albo i -rolami;

- b) \mathcal{J} spełnia aksjomat postaci $X_i \sqsubseteq Y_i$, co jest zapisywane $\mathcal{J} \models X_i \sqsubseteq Y_i$, wtwg $X_i^{I_i} \subseteq Y_i^{I_i}$;
- 2) spełnia połączony Lbox $\mathbf{L} = \{L_{ij} \mid i \neq j \in I\}$, co jest zapisywane $\mathcal{J} \models \mathbf{L}$ i oznacza, że spełnia każdy $L_{ij} \in \mathbf{L}$:
- a) \mathcal{J} spełnia $L_{ij} \in \mathbf{L}$, co jest zapisywane $\mathcal{J} \models L_{ij}$ wtwg spełnia każdy aksjomat postaci $p_{ij} \sqsubseteq q_{ij}$ z L_{ij} , gdzie p_{ij}, q_{ij} są ij -relacjami;
- b) \mathcal{J} spełnia aksjomat postaci $p_{ij} \sqsubseteq q_{ij}$, co jest zapisywane $\mathcal{J} \models p_{ij} \sqsubseteq q_{ij}$, wtwg $p_{ij}^{I_{ij}} \subseteq q_{ij}^{I_{ij}}$;
- 3) spełnia połączony Abox $\mathbf{A} = \{A_i \mid i \in I\}$, co jest zapisywane $\mathcal{J} \models \mathbf{A}$ i oznacza, że spełnia każdy $A_i \in \mathbf{A}$:
- a) \mathcal{J} spełnia $A_i \in \mathbf{A}$, co jest zapisywane $\mathcal{J} \models A_i$, wtwg spełnia każdą asercję postaci $C_i(a_i), R_i(a_i, b_i), p_{ij}(a_i, x_j)$ lub $q_{ji}(x_j, a_i)$ z A_i , gdzie C_i jest i -konceptem, R_i jest i -rolą, p_{ij} jest ij -relacją, q_{ji} jest ji -relacją, a_i, b_i są i -osobnikami, a x_j jest j -osobnikiem;
- b) \mathcal{J} spełnia asercję postaci $C_i(a_i)$ (lub $R_i(a_i, b_i), p_{ij}(a_i, x_j), q_{ji}(x_j, a_i)$), co jest zapisywane $\mathcal{J} \models C_i(a_i)$ (lub odpowiednio $\mathcal{J} \models R_i(a_i, b_i), \mathcal{J} \models p_{ij}(a_i, x_j), \mathcal{J} \models q_{ji}(x_j, a_i)$), wtwg $a_i^{I_i} \in C_i^{I_i}$; (lub odpowiednio pary elementów dziedzin $(a_i^{I_i}, b_i^{I_i}) \in R_i^{I_i}, (a_i^{I_i}, x_j^{I_j}) \in p_{ij}^{I_{ij}}, (x_j^{I_j}, a_i^{I_i}) \in q_{ji}^{I_{ji}}$).

■

Technika \mathcal{E} -connections jest w swojej istocie bardzo podobna do techniki DDL. W obu chodzi o to, aby odpowiednie konstrukcje nałożone na dwie ontologie pozwoliły ograniczyć wnioski możliwe do wyciągnięcia w ontologii docelowej za pomocą wniosków możliwych do wyciągnięcia w ontologii źródłowej. W ten sposób można wykorzystać relację subsumcji. Jeśli np. w j -tej ontologii zdefiniowano aksjomat $D \sqsubseteq C$, a w i -tej ontologii aksjomaty $G \equiv \exists p.C$ i $H \equiv \exists p.D$, gdzie $p \in \mathcal{E}_{ij}$, to konsekwencją logiczną jest zdanie $H \sqsubseteq G$. Dokładniej zobrazuje to przykład 4.5.

Przykład 4.5 (projekty medyczne w systemie \mathcal{E} -Connections)

Rozpatrzmy ponownie układ z projektami medycznymi, znany z przykładów 4.2 (str. 57), 4.3 (str. 62) i 4.4 (str. 68). Treść aksjomatów obu łączonych modułów jest początkowo taka sama. Moduł *projekty* (tu również oznaczony symbolem 1) zawiera następujące aksjomaty:

$$1:(\text{Projekt_dot_schorzeń_genetycznych} \equiv \text{Projekt} \sqcap \exists \text{dotyczy.Schorzenia_genetyczne}), \quad (4.32)$$

$$1:(\text{ProjektUE_dot_mukowiscydozy} \equiv \text{ProjektUE} \sqcap \exists \text{dotyczy.Mukowiscydoza}), \quad (4.33)$$

$$1:(\text{ProjektUE} \sqsubseteq \text{Projekt}), \quad (4.34)$$

$$1:(\exists \text{dotyczy.T} \sqsubseteq \text{Projekt}). \quad (4.35)$$

Moduł *schorzenia* (2) zawiera następujące aksjomaty:

$$2:(\text{Zwłóknienie_genet} \equiv \text{Zwłóknienie} \sqcap \exists \text{spowodowane_przez.Przyczyny_genet}), \quad (4.36)$$

$$2:(\text{Mukowiscydoza} \sqsubseteq \text{Zwłóknienie_genet} \sqcap \exists \text{zlokalizowane_w.Trzustka}). \quad (4.37)$$

Nie zmienia się także formuła, którą należy udowodnić:

$$1:(\text{ProjektUE_dot_mukowiscydozy} \sqsubseteq \text{Projekt_dot_schorzeń_genetycznych}). \quad (4.38)$$

Technologia \mathcal{E} -Connections nakazuje w miejsce reguł pomostowych zdefiniować relacje łączące $\mathcal{E}_{12} = \{spowodowane_przez, identyczny_z\}$ oraz dodatkowe aksjomaty w module *projekty*:

$$1:(\exists identyczny_z.\exists spowodowane_przez.Przyczyny_genet \sqsubseteq \text{Schorzenie_genetyczne}), \quad (4.39)$$

$$1:(Mukowiscydoza \sqsubseteq \exists identyczny_z.Mukowiscydoza) \quad (4.40)$$

Aksjomaty te wprowadzają do tego modułu relację łączącą (czyli 12-relację) o nazwie *identyczny_z*. Tworzy ona związki między konceptami z modułu 1, *Mukowiscydoza* i *Schorzenie_genetyczne*, a konceptami z modułu 2, odpowiednio: *Mukowiscydoza* i *∃spowodowane_przez.Przyczyny_genet*. Reguły te pełnią tę samą rolę, co reguły pomostowe w przykładzie 4.4, czyli wymuszają, aby algorytm *tableau* stwierdził, że w module *projekty* koncept *Mukowiscydoza* jest subsumowany przez koncept *Schorzenie_genetyczne*. ■

4.3 Podsumowanie

W niniejszym rozdziale naszkicowano szereg zagadnień związanych z wprowadzaniem pojęcia kontekstu do różnych formalizmów logicznych. Początkowo główną motywacją prowadzonych w tym zakresie prac była chęć odwzorowania pewnych zjawisk obecnych w języku naturalnym, istotnych z punktu widzenia wartościowania niektórych typów zdań. Z czasem motywacja została rozszerzona o problemy związane z postawami propozycjonalnymi i z różnymi kwestiami natury epistemicznej. Dzięki podjętym pracom możliwe stało się określenie pewnych zasad, które powinny być spełnione przez te systemy, które zostały stworzone w celu umożliwienia kontekstowego ujmowania wiedzy. Zasady te, opisane w punkcie 4.1.1, stanowią podstawę wielu głównych wątków niniejszej rozprawy, przede wszystkim wytyczają kryteria, według których da się oceniać proponowane formalizmy.

W pozostałych punktach podrozdziału 4.1 zamieszczono opis kilku formalizmów rozwijających logikę zdaniową lub logikę pierwszego rzędu w kierunku kontekstowości. Przedstawiono logikę zaimków wskazujących LD, kontekstową logikę zdań PLC oraz system LMS/MCS.

W praktycznych badaniach prowadzonych w ramach inicjatywy Semantic Web przyjęło się podejście kontekstowe wiązać z opracowaniami dotyczącymi sposobów modularyzacji ontologii. Podrozdział 4.2 przedstawia różne technologie modularne, przede wszystkim dwie najważniejsze i najlepiej dopracowane metody tworzenia ontologicznych systemów kontekstowych: DDL i \mathcal{E} -Connections. Technologie te stanowią praktyczne zastosowanie formalizmu LMS/MCS i jako takie stanowią interesujące studium kontekstowego ujmowania wiedzy w warunkach praktycznych. Istnieje przekonanie, że metody te są w stanie rozwiązać w zasadzie wszystkie wymienione we wstępie do niniejszej pracy problemy referencyjne. Problem integracyjny jest rozwiązany siłą rzeczy, gdyż właśnie integracja różnych modułów jest istotą tych technologii. Problem konstrukcyjny i organizacyjny jest rozwiązany dzięki temu, że będzie można od razu podzielić tworzoną ontologię na nieduże, łatwe do zrozumienia fragmenty i rozdzielić prace między różne zespoły projektowe. Modularność zwiększy też wydajność wnioskowania i umożliwi przydzielanie użytkownikom uprawnień. Tak więc zostaną również rozwiązane problemy wydajnościowy i ochrony danych.

Przekonanie o tym, że metody te rozwiązują problemy referencyjne sprawia, że są one traktowane jako ważne kierunki prac związanych z upowszechnieniem semantycznego opisu

informacji zawartych w Internecie. Sprawia też, że są one tymi rozwiązaniami, z którymi przede wszystkim musi być konfrontowana metoda kontekstualizacji proponowana w niniejszej pracy. Należy zatem więcej miejsca poświęcić na ich ocenę w świetle celów ujętych w tezach rozprawy.

Metody DDL i \mathcal{E} -Connections wciąż są rozwijane, ale skala ich zastosowania nie jest duża. Trudno więc powiedzieć, czy i w jakim stopniu spełniają postawione przed nimi cele. Z dostępnych opracowań wynika, że ich autorzy koncentrują się na zadaniu łączenia ontologii już istniejących w Internecie, chodzi więc przede wszystkim o rozwiązanie problemu integracyjnego. Zadanie to najbardziej kojarzy się z dziedziną nazywaną łączeniem ontologii (ang. *ontology merging*) ([N2004]). Nie jest to jednak rozwijanie idei kontekstowego ujmowania wiedzy. Zdaniem autora niniejszej rozprawy rozwiązania te, podobnie jak inne przedstawione w tym rozdziale, bazują na błędnym założeniu. W systemie LMS/MCS powiązania między modułami nie mają znaczenia semantycznego. Stanowiące je reguły pomostowe są jedynie zapisem sposobu przeniesienia wiedzy zawartej w jednym module do wnętrza drugiego modułu. Podobnie jest w systemie PLC, przy czym zastosowano tam nieco inną technikę. Jeśli z systemie LMS/MCS elementami służącymi do przeniesienia wiedzy są reguły wnioskowania, to w systemie PLC służą temu celowi formuły rozproszone w różnych modułach. Wynikiem takiego działania nie będzie utworzenie struktury kontekstów, w ramach której miejsce będzie określało znaczenie kontekstu, lecz utworzenie dość przypadkowego zbioru modułów będących *de facto* zbiorami formuł dość przypadkowo rozdzielonych i kojarzonych za pomocą kolejnych zbiorów formuł. W najlepszym przypadku można taki twór porównać do nieuporządkowanej sieci modułów, w której położenie danego węzła względem innych jej węzłów nie ma żadnego znaczenia z punktu widzenia modelowanego problemu. W przypadku dużych ontologii modułarnych, wynikiem takiego podziału może być jeszcze większy chaos. Pojawiają się pytania: Gdzie co znaleźć? Gdzie co umieścić? W jaki sposób dane zagadnienia rozdzielić? Do poważniejszych wątpliwości może dojść w sytuacji, gdy trzeba będzie dokonać w ontologii zmian. Nawet jeśli zespół tworzący taką ontologię będzie bardzo dobrze zorganizowany, jeśli dokumentacja będzie wykonana bardzo starannie brak semantycznych zależności między modułami spowoduje, że złagodzenie problemów konstrukcyjnego i organizacyjnego będzie tylko częściowe.

Najważniejszym brakiem opisywanych metod jest to, że nie rozwiązują one problemów wskazanych we wstępie rozprawy i umieszczonych w drugiej tezie pomocniczej: problemu adekwatności kontekstów i problemu projektowania kontekstowego. Problemy te powinny zostać rozwiązane od strony semantycznej, a nie syntaktycznej, na której technologii te się koncentrują. Z tego też powodu ich przydatność do organizowania kontekstowych baz wiedzy jest ograniczona.

5 Metoda SIM hierarchicznej kontekstualizacji ontologii

Z przedstawionych w podrozdziałach 3.3 i 4.3 ocen metod normalizacji i modularyzacji wynika, że żadna z nich nie spełnia w pełni pokładanych w nich nadziei. Metody normalizacji dostarczają bardzo przydatnych sposobów uporządkowania procesu konceptualizacji, ale w niewielkim stopniu niwelują niektóre problemy referencyjne (integracyjny i konstrukcyjny). W szczególności nie są w stanie ograniczyć problemu wydajnościowego ani problemu organizacyjnego. Z kolei metody modularyzacji nieźle radzą sobie z problemem integracyjnym, ale w zasadzie nie wypowiadają się o tym, w jaki sposób projektować kontekstowe bazy wiedzy, aby rozwiązać problem konstrukcyjny i organizacyjny. Autor niniejszej rozprawy jest przekonany, że aby osiągnąć pozytywny efekt, należy połączyć potencjał tkwiący w obu ideach. Z idei modularyzacji należy zaczerpnąć wszystko, co wiąże się ze sposobem, w jaki kontekstowość wpływa na treść posiadanej informacji, zaś z idei normalizacji wziąć wiedzę o przebiegu procesu konceptualizacji. Pomysł, którego obecnie brakuje, a który jest potrzebny, aby połączyć obie idee, to koncepcja budulca, z którego można byłoby składać elementy systemu modularnego i który mógłby być elementem brany pod uwagę w trakcie dokonywania konceptualizacji. Z technicznego punktu widzenia budulec powinien mieć cechy pozwalające tworzyć moduły i ograniczenia strukturalne opisujące przepływ wniosków. Z logicznego punktu widzenia powinien posiadać cechy pozwalające użyć go jako środka do wyrażenia złożonych zależności pomiędzy światami możliwymi dostrzeganymi przez twórców ontologii.

Prezentowana w tym rozdziale metoda SIM (opisywana także między innymi w publikacjach [GWW2007a], [GWW2007b], [GW2008a] i [GW2008b]) dostarcza takiego właśnie budulca. W podrozdziale 5.1 zdefiniowane są podstawowe elementy składające się na ten budulec. Tworzą one strukturę hierarchicznych ontologii kontekstowych. Semantyka tych struktur jest zgodna z wizją konceptualizacji opisaną w punkcie 4.1.1. Dzięki tej semantyce, opisaną w podrozdziale 5.2, możliwe jest objęcie struktury kontekstowej bazy wiedzy procesem konceptualizacji. Podrozdział 5.3 opisuje problemy związane z wnioskowaniem w bazie zbudowanej zgodnie z metodą SIM. Podrozdział 5.4 zawiera podsumowanie.

5.1 Pojęcia związane z kontekstualizacją hierarchiczną

Podrozdział ten zawiera definicje wszystkich elementów, które składają na strukturę kontekstowej bazy wiedzy. Podstawowymi elementami tej struktury są typy i wystąpienia kontekstów, które pozwalają dokonać podziału bazy na fragmenty dowolnej wielkości. Zawartością tych elementów będą zdania jakiejś teorii logicznej. W niniejszej rozprawie przyjmuje się, że zastosowaną tu logiką jest logika opisowa. Elementy te są połączone w jedną całość za pomocą trzech typów relacji.

Struktura kontekstowej bazy wiedzy jest definiowana w następujący sposób:

Definicja 5.1 (struktura kontekstowej bazy wiedzy)

Strukturą kontekstowej bazy wiedzy jest struktura $S_{\mathcal{K}} = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \preceq, \ll, f_{inst})$, gdzie:

- 1) $\mathbf{T} \uplus \mathbf{A}$, to suma rozłączna zbioru typów kontekstów i zbioru wystąpień kontekstów;
- 2) relacja dziedziczenia $\preceq \subseteq \mathbf{T} \times \mathbf{T}$ jest (silnym) porządkiem częściowym nałożonym na zbiór \mathbf{T} ; relacja ta posiada element maksymalny T_{max} ;
- 3) relacja agregacji $\ll \subseteq \mathbf{A} \times \mathbf{A}$ jest (silnym) porządkiem częściowym nałożonym na zbiór \mathbf{A} ; relacja ta posiada element maksymalny A_{max} .

- 4) *funkcja konkretyzacji* $f_{inst}: \mathbf{A} \rightarrow \mathbf{T}$ jest funkcją przyporządkowującą każdemu wystąpieniu kontekstu typ; A jest *wystąpieniem kontekstu* T , jeżeli $f_{inst}(A) = T$. Funkcja f_{inst} ma następujące właściwości:
- f_{inst} jest funkcją całkowitą (każde wystąpienie jest określonego typu),
 - f_{inst} jest suriekcją (każdy typ ma przynajmniej jedno wystąpienie),
 - $f_{inst}(A_{max}) = T_{max}$,
 - jeżeli $A \ll A'$, to $f_{inst}(A) \leq f_{inst}(A')$.

Dla składników struktury przyjmuje się następujące zasady nazewnictwa:

- \mathbf{T} nazywa się *kontekstową terminologią*, a $T \in \mathbf{T}$ *typem kontekstu*,
- \mathbf{A} nazywa się *kontekstowym opisem świata*, a $A \in \mathbf{A}$ *wystąpieniem kontekstu*,
- T_{max} jest *typem podstawowym* kontekstowej terminologii \mathbf{T} ,
- T *dziedziczy od (jest potomkiem)* T' , gdzie $T, T' \in \mathbf{T}$, jeżeli $T \leq T'$,
- T *jest dziedziczony przez (jest przodkiem)* T' , gdzie $T, T' \in \mathbf{T}$, jeżeli $T' \leq T$.
- A_{max} jest *wystąpieniem podstawowym* kontekstowego opisu świata \mathbf{A} ,
- A *agreguje* A' , gdzie $A, A' \in \mathbf{A}$, jeżeli $A' \ll A$,
- A *jest agregowane przez* A' , gdzie $A, A' \in \mathbf{A}$, jeżeli $A \ll A'$,
- $A \in \mathbf{A}$ jest *wystąpieniem agregującym*, nazywanym też *węzłem agregacji*, jeżeli $\exists A' \in \mathbf{A} (A' \ll A)$,
- $A \in \mathbf{A}$ jest *wystąpieniem agregowanym*, jeżeli $\exists A' \in \mathbf{A} (A \ll A')$.

Zdefiniowana powyżej struktura jest głównym składnikiem kontekstowej bazy wiedzy. W ramach tej struktury umieszcza się ontologię podzieloną na odpowiednie fragmenty. Ontologię tę nazywa się *zawartością bazy wiedzy*, a jej fragmenty są przyporządkowywane poszczególnym elementom struktury za pomocą funkcji zawartości f_{cont} :

Definicja 5.2 (funkcja zawartości struktury kontekstowej bazy wiedzy)

Funkcją zawartości f_{cont} *struktury kontekstowej bazy wiedzy* $S_{\mathcal{K}} = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \leq, \ll, f_{inst})$ oraz ustalonej logiki opisowej \mathcal{L} jest funkcja całkowita $f_{cont}: \mathbf{T} \uplus \mathbf{A} \rightarrow \mathcal{P}(\mathcal{L})$, która przypisuje każdemu typowi i każdemu wystąpieniu kontekstu odpowiednie zbiory zdań zbudowane zgodnie z formalizmem logiki \mathcal{L} :

- $\forall T \in \mathbf{T}$ zachodzi $f_{cont}(T) \in \mathcal{P}(\mathcal{L}_T)$,
- $\forall A \in \mathbf{A}$ zachodzi $f_{cont}(A) \in \mathcal{P}(\mathcal{L}_A)$.

$f_{cont}(T)$ nazywa się *zawartością typu* T , a $f_{cont}(A)$ *zawartością wystąpienia* A .

Definicją typu T nazywa się jego zawartość zsumowaną z zawartością wszystkich typów dziedziczonych: $def(T) = f_{cont}(T) \cup \bigcup_{T' \in \mathbf{T} T \leq T'} f_{cont}(T')$.

Dodatkowo funkcja f_{cont} ma następującą właściwość:

- zawartość wystąpienia kontekstu nie może wprowadzać nowych nazw konceptów i ról; dopuszczalne są jedynie te, które są zdefiniowane w ramach definicji typu:
 $\forall A \in \mathbf{A} (Sig_C(f_{cont}(A)) \subseteq Sig_C(def(f_{inst}(A))) \wedge Sig_R(f_{cont}(A)) \subseteq Sig_R(def(f_{inst}(A))))$

Wymagania dotyczące funkcji zawartości bazy wiedzy, mówiące że typom kontekstów można przypisywać tylko aksjomaty, a wystąpieniom tylko asercje jest zrozumiałe i zgodne z intuicją. Są to przecież elementy odpowiadające Tboxom i Aboxom ontologii. Dodatkowym, bardzo ważnym ograniczeniem jest właściwość wymieniona w punkcie 3), według której zawartość wystąpienia kontekstu nie może wprowadzać do słownika własnych nazw konceptów i ról.

Kontekstowa baza wiedzy jest złożeniem struktury i funkcji zawartości:

Definicja 5.3 (kontekstowa baza wiedzy)

Kontekstową bazą wiedzy \mathcal{K} dla ustalonej logiki opisowej \mathcal{L} jest para: $\mathcal{K} = (S_{\mathcal{K}}, f_{cont})$, gdzie $S_{\mathcal{K}}$ jest strukturą kontekstowej bazy wiedzy, a f_{cont} funkcją zawartości tej struktury. ■

W dalszym ciągu rozważań w celu uproszczenia zapisu będą jeszcze używane następujące pojęcia:

Definicja 5.4 (agregacja bezpośrednia, dziedziczenie bezpośrednie)

Agregacją bezpośrednią $\ll\bullet$ nazywa się podzbiór relacji agregacji, taki że $A \ll\bullet A'$ wtwg $A \ll A' \wedge \neg\exists A'' (A \ll A'' \wedge A'' \ll A')$.

Dziedziczeniem bezpośrednim $\preceq\bullet$ nazywa się podzbiór relacji dziedziczenia, taki że $T \preceq\bullet T'$ wtwg $T \preceq T' \wedge \neg\exists T'' (T \preceq T'' \wedge T'' \preceq T')$. ■

Definicja 5.5 (ścieżka dziedziczenia)

Ścieżką dziedziczenia między typami T i T' , co jest zapisywane $\preceq_{T-T'}$, jest dowolny z ciągów (T_1, \dots, T_n) , taki że $T = T_1 \wedge T_n = T' \wedge ((n \geq 2) \rightarrow (\forall i \in [1, n-1] (T_i \preceq\bullet T_{i+1})))$. n jest długością ścieżki dziedziczenia.

Ścieżką dziedziczenia dla typu T jest $\preceq_{T-T_{max}}$. ■

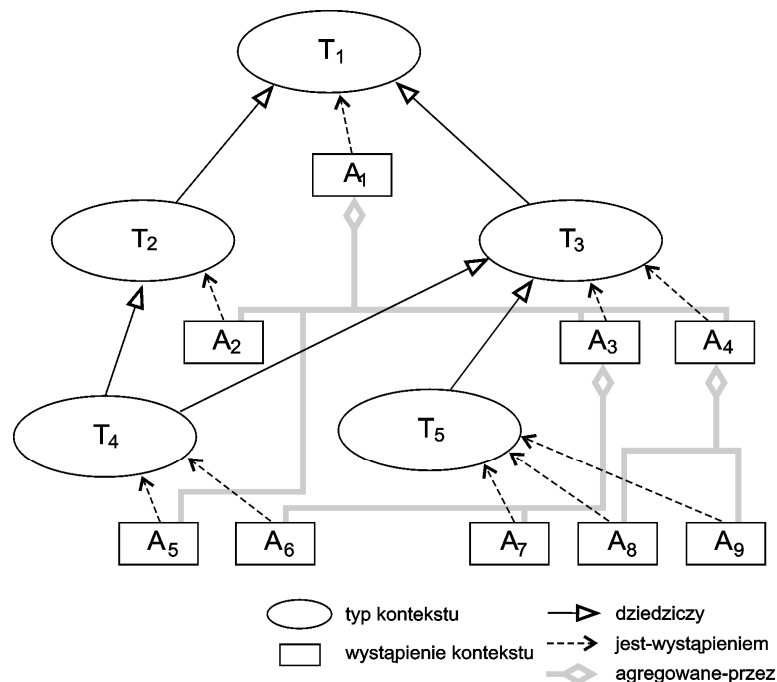
Cechą charakterystyczną kontekstowej bazy wiedzy jest to, że obie części ontologii — terminologia i opis świata — są od siebie, w porównaniu z innymi metodami modularyzacji, bardziej niezależne. Nie jest już tak, że modułami są pary Tbox-Abox. Funkcja f_{inst} pozwala łączyć wiele opisów świata z jedną terminologią. Obie części ontologii są porządkowane odrębnymi relacjami — terminologie relacją dziedziczenia, a opisy świata relacją agregacji. Nazwa „relacja dziedziczenia” jest dość oczywista, gdyż odzwierciedla fakt, iż potomek dziedziczy wszystkie cechy przodka. Nazwa „relacja agregacji” wynika stąd, że wystąpienie agregujące łączy informacje z wystąpień agregowanych (patrz definicje 5.7 i 5.10).

Pomiędzy elementami obu części istnieje powiązanie opisane funkcją konkretyzacji, która przypisuje każdemu wystąpieniu typ. Główną właściwością tej funkcji jest możliwość powiązania wielu wystąpień z danym typem kontekstu. Możliwość ta, połączona z możliwościami dostarczonymi przez relację agregacji, tworzy podstawy kontekstowego opisu świata, czyli ujmowania różnych zjawisk z różnych punktów widzenia. W skrajnych przypadkach możliwe jest nawet istnienie wystąpień kontekstów, które przechowują informacje sprzeczne. Warunkiem jest jedynie, aby nie pojawiły się one w kontekście agregującym. Ten problem będzie omawiany w dalszej części niniejszej rozprawy.

Przykładową ontologię skonstruowaną zgodnie z zasadami metody SIM demonstruje rys. 5.1. Ontologia ta składa się z pięciu typów kontekstów ($\mathbf{T} = \{T_i \mid i \in [1, 5]\}$, $T_{max} = T_1$) i dziewięciu wystąpień ($\mathbf{A} = \{A_j \mid j \in [1, 9]\}$, $A_{max} = A_1$). Niektóre z typów mają więcej niż jedno wystąpienie, na przykład wystąpienia A_7, A_8 i A_9 konkretyzują typ T_5 .

Warto zauważyć, że w kontekstowym opisie świata znajduje się jedno wystąpienie kontekstu agregujące wszystkie pozostałe wystąpienia kontekstów. Jest to wystąpienie podstawowe, będące elementem maksymalnym relacji agregacji. Na omawianym rysunku jest nim A_1 . Wynika z tego, że wszystkie wystąpienia kontekstów muszą być spójne względem

siebie na najwyższym (zdefiniowanym w ramach kontekstowej terminologii) poziomie uogólnienia. Jest to też podstawą, aby parę złożoną z kontekstowego opisu świata i kontekstowej terminologii można było nazwać kontekstową bazą wiedzy.



Rys. 5.1 Przykład kontekstowej bazy wiedzy zaprojektowanej wg metody SIM. Dla przejrzystości na rysunku pokazano tylko dziedziczenia i agregacje bezpośrednie.

Wymaganie, aby funkcja konkretyzacji była suriekcją, jest podyktowane sposobem, w jaki w dalszej części rozdziału zdefiniowana jest semantyka kontekstowej bazy wiedzy. W metodzie SIM wnioskowanie prowadzi się z punktu widzenia jednego kontekstu. Kontekst jest definiowany w sposób następujący:

Definicja 5.6 (kontekst)

Kontekst κ jest parą $\kappa = (T, A)$, gdzie T jest typem kontekstu, a A jednym z jego wystąpień. Terminologią kontekstu jest definicja jego typu $T(\kappa) = def(T)$. ■

Jak widać kontekst jest wyznaczany przez wystąpienie. Gdyby mogły istnieć typy kontekstów bez wystąpienia, to brak wystąpienia uniemożliwiłby wnioskowanie terminologiczne. Jeśli więc, modelując dziedzinę, projektant kontekstowej bazy wiedzy zadecyduje, że dany typ nie będzie miał wystąpień, to z teoretycznego punktu widzenia powstanie domyślne puste (nie zawierające asercji) wystąpienie agregowane przez wystąpienie podstawowe.

Metoda SIM nie wprowadza do syntaktyki logiki opisowej żadnych zmian. Jediną różnicą jest to, że nie tworzy się jednego zbioru zdań, lecz wiele zbiorów i przyporządkowuje je do poszczególnych elementów struktury za pomocą funkcji f_{cont} . Jednakże zawartość syntaktyczna nie jest tym samym, co opis. Na przykład typ kontekstu jest opisany nie tylko przez swoją zawartość, ale przez zawartość wszystkich typów dziedziczonych. Podobnie jest w przypadku wystąpień, jednak tu sytuacja jest nieco bardziej skomplikowana. Problem ten staje się widoczny już w definicji sygnatur związanych z kontekstem:

Definicja 5.7 (sygnatura typu i wystąpienia kontekstu)

Sygnaturą typu kontekstu T należącego do kontekstowej bazy wiedzy \mathcal{K} jest sygnatura jego definicji $S(T) = \text{Sig}(\text{def}(T))$.

Poszczególne składniki tej sygnatury zawierające nazwy konceptów, ról i osobników, to odpowiednio: $S_C(T) = \text{Sig}_C(\text{def}(T))$, $S_R(T) = \text{Sig}_R(\text{def}(T))$, $S_A(T) = \text{Sig}_A(\text{def}(T))$.

Sygnaturą wystąpienia kontekstu A należącego do kontekstowej bazy wiedzy \mathcal{K} nazywa się sygnaturę $S(A)$, która jest utworzona w następujący sposób:

$$S(A) = \text{Sig}_A(f_{\text{cont}}(A)) \cup S_A(f_{\text{inst}}(A)) \cup \bigcup_{A' \in \mathbf{A}: A' \ll A} (\text{Sig}_A(f_{\text{cont}}(A')) \cup S_A(f_{\text{inst}}(A'))). \quad \blacksquare$$

Typ kontekstu T jest zdefiniowany za pomocą aksjomatów umieszczonych w T i wszystkich jego przodkach. Import terminów pozalogicznych przebiega w kierunku od elementu „większego” do elementu „mniejszego”. Z kolei w przypadku wystąpień jest odwrotnie: wystąpienie agregujące importuje nazwy osobników z wystąpień agregowanych, czyli przepływ nazw przebiega od elementu „mniejszego” do elementu „większego”. Uzasadnia to przyjęcie określenia „agregacja” dla relacji wiążącej wystąpienia.

Definicja 5.8 (sygnatura kontekstu)

Sygnaturą kontekstu $\kappa = (T, A)$ nazywamy sumę $S(\kappa) = S(T) \cup S(A)$. Poszczególne składniki tej sygnatury zawierające nazwy konceptów, ról i osobników, to odpowiednio:

$$S_C(\kappa) = S_C(T),$$

$$S_R(\kappa) = S_R(T),$$

$$S_A(\kappa) = S(A). \quad \blacksquare$$

Sygnatura kontekstu zawiera sygnaturę typu i wystąpienia. Kontekst jest, jak to już zostało powiedziane w komentarzu do definicji 5.6, miejscem wnioskowania o świecie. Jednak to wnioskowanie nigdy nie odbywa się z uwzględnieniem całej posiadanej wiedzy. W danej sytuacji należy podjąć decyzję: czy wnioskować w kontekście „większym” (z punktu widzenia obu relacji), czyli dla większej liczby osobników, ale dla uboższej (bardziej ogólnej) terminologii, czy w kontekście „mniejszym”, czyli dysponując bardziej precyzyjnym słownikiem, ale na temat mniejszej liczby osobników.

Sama struktura kontekstowej bazy wiedzy jest opisywana zgodnie z zasadami teorii mnogości. Oto przykład takiego opisu dla małej ontologii.

Przykład 5.1 (Kontekstowice — zapis)

Przykładowa ontologia składa się z trzech typów kontekstów, z których każdy ma jedno wystąpienie (ilustracja tego przykładu znajduje się na rys. 5.2).

$$S_{\mathcal{K}} = (\{T_1, T_2, T_3\} \cup \{A_1, A_2, A_3\}, T_1, A_1, \{(T_2, T_1), (T_3, T_1)\}^*, \{(A_2, A_1), (A_3, A_1)\}^*, \{(A_1, T_1), (A_2, T_2), (A_3, T_3)\})$$

$$f_{\text{cont}}(T_1) = \{Mężczyzna \sqcap Kobieta \equiv \perp\}$$

$$f_{\text{cont}}(T_2) = \{Sopran \sqsubseteq Kobieta\}$$

$$f_{\text{cont}}(T_3) = \{Matka \equiv Rodzic \sqcap Kobieta\}$$

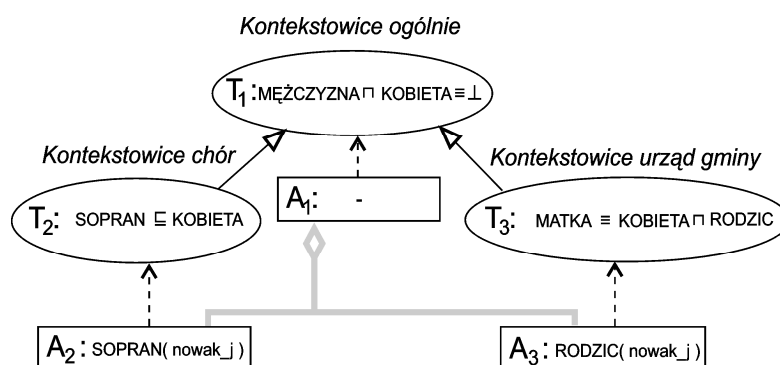
$$f_{\text{cont}}(A_1) = \{\}$$

$$f_{\text{cont}}(A_2) = \{Sopran(\text{nowak}_j)\}$$

$$f_{\text{cont}}(A_3) = \{Rodzic(\text{nowak}_j)\}$$

Dziedziną struktury $S_{\mathcal{K}}$ kontekstowej bazy wiedzy \mathcal{K} jest suma dwóch zbiorów, z których jeden zawiera typy T_1, T_2, T_3 , a drugi wystąpienia A_1, A_2, A_3 . Typem podstawowym jest T_1 , wystąpieniem podstawowym A_1 . Relacja dziedziczenia składa się z dwóch par, które mówią,

że zarówno typ T_2 , jak i typ T_3 dziedziczą od typu T_1 . Gwiazdka (*), która oznacza domknięcie przechodnie relacji, nie jest więc w tym przypadku potrzebna. Jednak dla zgodności z definicją struktury kontekstowej bazy wiedzy lepiej ją zawsze przy opisie umieścić.



Rys. 5.2 Ilustracja kontekstowej bazy wiedzy zdefiniowanej w przykładzie 5.1.

Relacja agregacji wskazuje, że wystąpienie A_1 agreguje wystąpienia A_2 i A_3 . Także i tutaj lepiej zaznaczyć domknięcie przechodnie. Funkcja konkretyzacji przyporządkowuje wystąpienia A_1, A_2, A_3 do typów T_1, T_2, T_3 . W kolejnych krokach opisana jest funkcja f_{cont} , czyli zawartości typów i wystąpień kontekstów. Każdy typ zawiera po jednym aksjomacie. Wystąpienia A_2 i A_3 zawierają po jednej asercji, a wystąpienie A_1 jest puste. ■

5.2 Semantyka ontologii hierarchicznej

Szczególną cechą metody SIM jest ustrukturalizowana interpretacja. Stąd właśnie nazwa tej metody. Interpretacja ta jest zdefiniowana w następujący sposób:

Definicja 5.9 (interpretacja kontekstowa)

Interpretacją kontekstową kontekstowej bazy wiedzy $\mathcal{K} = (S_{\mathcal{K}}, f_{cont})$, gdzie $S_{\mathcal{K}} = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \preceq, \ll, f_{inst})$, jest funkcja $\mathcal{J}: \mathbf{A} \rightarrow \bigcup_{A \in \mathbf{A}} \mathcal{I}^{\sigma}(S(A) \cup S(f_{inst}(A)))$, gdzie $\mathcal{I}^{\sigma}(S)$ jest zbiorem wszystkich interpretacji dla sygnatury S o postaci jak w definicji 2.9, przyporządkowująca każdemu wystąpieniu kontekstu interpretację lokalną. Funkcja \mathcal{J} ma następujące cechy:

- 1) $\forall A, A' \in \mathbf{A} (A \ll A' \rightarrow \Delta^{\mathcal{J}(A)} \subseteq \Delta^{\mathcal{J}(A')})$.
- 2) $\forall A, A' \in \mathbf{A} (\forall a \in S(A) \cap S(A') (a^{\mathcal{J}(A)} = a^{\mathcal{J}(A')}))$.

Interpretacja lokalna przyporządkowywana każdemu wystąpieniu kontekstu, podobnie, jak interpretacja standardowej ontologii wyrażonej w logice opisowej — patrz definicja 2.9, jest parą $(\Delta^{\mathcal{J}}, \bullet^{\mathcal{J}})$. Struktura interpretacji kontekstowej jest determinowana przez funkcję \mathcal{J} i relację agregacji.

Ograniczenia nałożone na dziedziny i nazwy osobników mogą utrudnić łączenie heterogenicznych baz wiedzy w jedną strukturę hierarchiczną. Jednak w sytuacji, gdy celem modularyzacji jest udoskonalenie procesu tworzenia jednorodnego modelu danej dziedziny, ograniczenie to nie powinno sprawiać kłopotów.

Definicja spełniania określa właściwości, jakie musi posiadać interpretacja, aby stać się modelem kontekstowej bazy wiedzy.

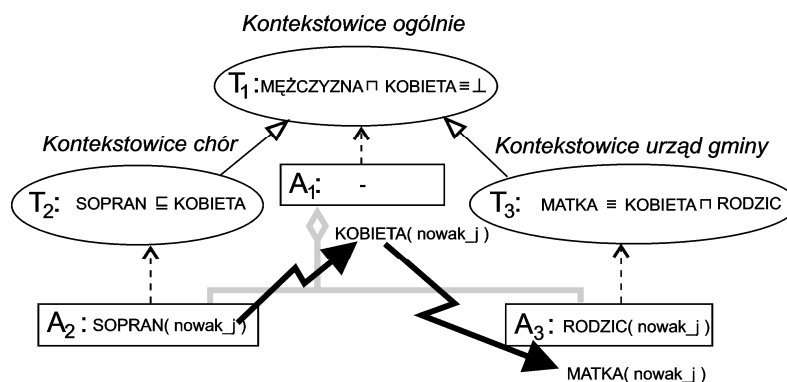
Definicja 5.10 (spełnianie)

Interpretacja kontekstowa \mathcal{J} spełnia kontekstową bazę wiedzy $\mathcal{K} = (S_{\mathcal{K}}, f_{cont})$, gdzie $S_{\mathcal{K}} = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \sqsubseteq, \ll, f_{inst})$, co jest zapisywane $\mathcal{J} \models \mathcal{K}$, wtwg dla $\forall A \in \mathbf{A}$ i $T = f_{inst}(A)$:

- 1) $\mathcal{J}(A) \models f_{cont}(A)$ w sensie podanym w definicji 2.10,
- 2) $\mathcal{J}(A) \models def(T)$ w sensie podanym w definicji 2.10,
- 3) $\forall A \ll A' (\forall C \in \mathcal{S}_C(f_{inst}(A'))) (C^{\mathcal{J}(A)} \cap \Delta^{\mathcal{J}(A)} = C^{\mathcal{J}(A')})$,
- 4) $\forall A \ll A' (\forall R \in \mathcal{S}_R(f_{inst}(A'))) (R^{\mathcal{J}(A)} \cap \Delta^{\mathcal{J}(A)} \times \Delta^{\mathcal{J}(A)} = R^{\mathcal{J}(A')})$.

Interpretacja, która spełnia kontekstową bazę wiedzy, jest nazywana jej *modelem*. ■

Zasady podane w punktach 1) i 2) definicji 5.10 nazywamy *zasadami spełniania lokalnego*. Zadaniem ich jest zapewnić, aby interpretacja spełniła lokalne zbiory formuł. W przypadku zasady 2), dotyczącej typów kontekstów, spełnianie dotyczy nie tylko lokalnie zdefiniowanych aksjomatów, lecz także wszystkich aksjomatów przypisanych typom dziedzicznym. Zasady podane w punktach 3) i 4) nazywają się *zasadami zachowania modelu*. Zapewniają one właściwość nazywaną *zachowawczym rozszerzeniem* (ang. *conservative extension*) (na przykład w [GLW2006], [CHKS2007]), rozwiniętą później w [GHKS2008] do *rozszerzenia zachowującego model* (ang. *model conservative extension*), które polega na tym, że jeśli nowa terminologia importuje i rozszerza terminologię już istniejącą, to nie może zmieniać interpretacji terminów w niej zdefiniowanych. Zachowanie modelu rozszerzanej ontologii jest warunkiem gwarantującym, że tworzona baza wiedzy pozostanie spójna. W przypadku metody SIM zasady zachowania modelu sterują przepływem wniosków między kontekstami. Sposób przepływu wniosków ilustruje kolejny przykład.



Rys. 5.3 Ilustracja przepływu wniosków w ramach kontekstowej bazy wiedzy opisanego w przykładzie 5.2.

Przykład 5.2 (Kontekstowice — przepływ wniosków)

Korzystamy w tym przykładzie z ontologii zdefiniowanej z przykładzie 5.1. Typ T_1 (*Kontekstowice ogólnie*) posiada jedynie aksjomat mówiący, że koncepty *Mężczyzna* i *Kobieta* są rozłączne. Typ T_2 (*Kontekstowice chór*) mógłby zawierać wiele aksjomatów opisujących koncepty, role i zależności między nimi w kontekście miejscowego chóru. Dla potrzeb przykładu istotny jest aksjomat *Sopran* \sqsubseteq *Kobieta* mówiący, że osoby posługujące się sopranem mogą być tylko kobietami. Typ T_3 (*Kontekstowice urząd gminy*) definiuje pojęcia z zakresu ról społecznych. Jednym z takich pojęć jest koncept *Matka* opisany jako wspólna część konceptu *Rodzic* i *Kobieta*. Załóżmy, że wnioskowanie odbywa się w kontekście $\kappa_3 = (T_3, A_3)$. W wystąpieniu A_3 jawnie zostało powiedziane, że osobnik

o nazwie *nowak_j* jest rodzicem. Jednakże we wszystkich kontekstach widoczny jest koncept *Kobieta*. Zasada zachowania modelu z punktu 3) definicji 5.10 mówi, że koncept ten we wszystkich wystąpieniach powiązanych relacją agregacji musi być interpretowany jednakowo w ramach ich dziedzin. Z kolei punkt 2) z definicji 5.9 mówi, że nazwy osobników muszą być jednakowo interpretowane w całej bazie wiedzy. W konsekwencji możliwy jest przepływ wniosku, że osobnik *nowak_j* jest wystąpieniem konceptu *Kobieta* z wystąpienia kontekstu A_2 do wystąpienia agregującego A_1 , a stąd do kontekstu κ_3 , w którym odbywa się wnioskowanie. Dzięki temu urzędnicy gminy Kontekstowice mogą łatwo wywnioskować, że pani Nowak jest matką i być może należy jej się jakieś wsparcie. ■

5.3 Wnioskowanie w kontekstowej bazie wiedzy

W tym podrozdziale zawarto opis zagadnień związanych z wnioskowaniem z kontekstowej bazy wiedzy zbudowanej zgodnie z metodą SIM. W poprzednich punktach przedstawiono ogólne zasady rządzące konstrukcją takich baz oraz przepływem wniosków wzdłuż relacji agregacji i dziedziczenia między kontekstami. Tu zagadnienia wnioskowania zostaną przedstawione od strony praktycznej. W punkcie 5.3.1 sformułowana zostanie lista problemów wnioskowania i omówiony będzie jej związek z problemami wnioskowania z tradycyjnej bazy wiedzy wyrażonej w logice opisowej. W punktach 5.3.2 i 5.3.3 opisana zostanie budowa przykładowych algorytmów wnioskowania oraz kwestie związane z ich praktyczną implementacją.

5.3.1 Problemy wnioskowania kontekstowego

Sformalizowanie procesu wnioskowania wymaga określenia, czym jest wynikanie w kontekstowej bazie wiedzy.

Definicja 5.11 (wynikanie kontekstowe)

Niech $\mathcal{K} = (S_{\mathcal{K}}, f_{cont})$ będzie kontekstową bazą wiedzy, gdzie $S_{\mathcal{K}} = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \triangleq, \ll, f_{inst})$, zaś $T \in \mathbf{T}$ dowolnym wybranym typem kontekstu w tej bazie, a $A \in \mathbf{A}$ jednym z wystąpień tego typu. Mówi się, że:

- 1) aksjomat $C \sqsubseteq D$ wynika z \mathcal{K} w kontekście $\kappa = (T, A)$, co zapisuje się $\mathcal{K} \models_{\kappa} C \sqsubseteq D$, wtwg w każdej kontekstowej interpretacji \mathcal{J} będącej modelem \mathcal{K} prawdą jest, że $\mathcal{J}(A) \models C \sqsubseteq D$ (w sensie podanym w definicji 2.10),
- 2) asercja $C(a)$ (i analogicznie $R(a, b)$) wynika z \mathcal{K} w kontekście $\kappa = (T, A)$, co zapisuje się $\mathcal{K} \models_{\kappa} C(a)$ (i odpowiednio $\mathcal{K} \models_{\kappa} R(a, b)$), wtwg w każdej kontekstowej interpretacji \mathcal{J} będącej modelem \mathcal{K} prawdą jest, że $\mathcal{J}(A) \models C(a)$ (i odpowiednio $\mathcal{J}(A) \models R(a, b)$) (w sensie podanym w definicji 2.10). ■

Powyższa definicja jest ścisła, chociaż może być trochę myląca. Należy rozpatrywać ją łącznie z definicjami interpretacji kontekstowej (5.9) i spełniania (5.10) oraz definicją sygnatury (5.7). Definicje te określają sposób, w jaki metoda SIM realizuje postulaty zawarte w regułach lokalności i kompatybilności z rozdziału 4.1.1. Zgodnie z ujęciem przedstawionym w tamtym rozdziale każdy kontekst κ jest systemem aksjomatycznym o postaci $Th = (L, Ak, R)$. Można zapisać, że kontekst $\kappa = (L_{\kappa}, Ak_{\kappa}, R_{\kappa})$. Językiem kontekstu κ jest język L_{κ} . Dla uproszczenia przyjęto w tej rozprawie, że języki wszystkich kontekstów danej ontologii zbudowanej zgodnie z zasadami metody SIM bazują na tej samej logice opisowej, ale różnią się nie tylko słownikami symboli pozalogicznych, czyli sygnaturami, ale również

sposobem przeprowadzonej konceptualizacji¹⁷. Ak_κ jest zbiorem aksjomatów teorii, przy czym z punktu widzenia definicji systemu aksjomatycznego zbiorem tym jest suma zdań opisujących typ i wystąpienie tego kontekstu $f_{cont}(T) \cup f_{cont}(A)$. W końcu R_κ jest zbiorem reguł dowodzenia, które są, przy założeniu jednolitej logiki, takie same w każdym z kontekstów. Lokalność wnioskowania jest zachowana w sposób oczywisty. Zapewniają ją zasady spełniania lokalnego z definicji 5.10. Ograniczenia strukturalne SR są w metodzie SIM realizowane poprzez relacje dziedziczenia i agregacji. Umożliwiają one przepływ wniosków między kontekstami. Zasady zachowania modelu z definicji 5.10 opisują w sposób ogólny sposób działania tych ograniczeń. W tym podrozdziale jest to opisane bardziej szczegółowo.

Definicja 5.11 pokazuje, jak wynikanie jest rozumiane w bazie kontekstowej. Analiza tej definicji pozwala stwierdzić, że wnioskowanie z bazy wiedzy zgodnej z metodą SIM ma charakter lokalny i jest zawsze przeprowadzane z punktu widzenia jednego, wybranego kontekstu. Choć w procesie wnioskowania brane są pod uwagę informacje pochodzące z innych kontekstów, zawsze ich postać dostosowywana jest do specyfiki tego kontekstu, z punktu widzenia którego przeprowadzane jest wnioskowanie.

Przy opisanym wyżej założeniu, postać definicji 5.11 pozwala na prostą adaptację typowych problemów wnioskowania do specyfiki bazy SIM. Za standardowe problemy wnioskowania terminologicznego dla bazy kontekstowej \mathcal{K} można zatem uznać:

- problem podrzędności konceptów C i D w kontekście κ : tj. czy $\mathcal{K} \models_\kappa C \sqsubseteq D$,
- problem równoważności konceptów C i D w kontekście κ : tj. czy $\mathcal{K} \models_\kappa C \sqsubseteq D$ i $\mathcal{K} \models_\kappa D \sqsubseteq C$,
- problem rozłączności konceptów C i D w kontekście κ : tj. czy $\mathcal{K} \models_\kappa C \sqcap D \sqsubseteq \perp$,
- problem spełnialności konceptu C w kontekście κ : tj. czy $\mathcal{K} \models_\kappa C \sqsubseteq \perp$.

Z kolei standardowe problemy wnioskowania z opisu świata bazy kontekstowej \mathcal{K} obejmują:

- problem przynależności osobnika a do (ekstensji) konceptu C w κ : tj. czy $\mathcal{K} \models_\kappa C(a)$,
- problem przynależności pary osobników (a, b) do (ekstensji) roli R w κ : tj. czy $\mathcal{K} \models_\kappa R(a, b)$,
- problem wyspecyfikowania listy osobników, które na pewno należą do (ekstensji) konceptu C w κ : tj. podanie wszystkich a , takich że $\mathcal{K} \models_\kappa C(a)$.

Standardowe problemy wnioskowania są niewątpliwie bardzo istotne i pozwalają użytkownikowi na uzyskanie ważnych informacji. Jednak specyficzna struktura bazy SIM umożliwia sformułowanie nowych problemów, w szerszy sposób wykorzystujących hierarchię typów i wystąpień kontekstów. Poniższa lista wymienia kilka ciekawych niestandardowych rodzajów wnioskowania, na jakie pozwala struktura kontekstów:

- **odnalezienie kontekstu dla warunku**, czyli problem wyznaczenia zbioru kontekstów, dla którego spełniony jest pewien warunek, np.:
 - koncept C należy do sygnatury kontekstu,
 - osobnik a należy do sygnatury kontekstu,
 - dwa koncepty C i D są względem siebie podrzędne, równoważne lub rozłączne,

¹⁷ Więcej na temat możliwości zróżnicowania konceptualizacji w zależności od kontekstu — w rozdziale 6.

- koncept C jest spełnialny,
 - osobnik a należy do (ekstensji) konceptu C ,
 - para osobników (a, b) należy do (ekstensji) roli R ,
- **wnioskowanie z części bazy**, czyli rozwiązanie któregoś ze standardowych problemów wnioskowania, ale przy założeniu, że brane są pod uwagę tylko informacje jedynie z niektórych kontekstów. Wykorzystanie tego rodzaju wnioskowania pozwala na ustalenie pewnych dodatkowych atrybutów kontekstów, jak np. wiarygodność (wyrażana pewną miarą liczbową) i w konkretnych przypadkach wyciąganie wniosków jedynie z zadanej części bazy (na przykład tej, dla której wartość miary wiarygodności przekracza pewien próg),
 - **wyrażanie konceptu spoza kontekstu** w terminach właściwych dla danego kontekstu. Funkcja ta pozwala m.in. na ocenę wpływu informacji zewnętrznej na zawartość danego kontekstu. Każdy kontekst reprezentuje pewien ustalony punkt widzenia i pewne założenia, zaś dzięki przetłumaczeniu konceptu można uzyskać wgląd w to, na który fragment dziedziny oddziałują inne konteksty. Formalnie problem ten polega na znalezieniu skończonego (w sensie skończoności konstruktora) konceptu C w κ , którego ekstensja najściślej odpowiada ekstensji konceptu C' w κ' . Problem ten można uściślić do dwóch wariantów: odnalezienia konceptu C_{min} o najmniejszej ekstensji pokrywającej ekstensję C (przyciętą do dziedziny κ) lub odnalezienia konceptu C_{max} o największej ekstensji zawartej w ekstensji C (również przyciętej do dziedziny κ).

Powyższy wykaz problemów nie wyczerpuje oczywiście wszystkich możliwości przeprowadzania niestandardowego wnioskowania w kontekstowej bazie wiedzy. Stanowi on jednak ciekawy punkt odniesienia dla dyskusji właściwości algorytmów wnioskowania. Możliwość adaptacji niektórych z nich do efektywnego rozwiązywania powyższych problemów wskazuje na dodatkową wartość takiego algorytmu w procesie wnioskowania kontekstowego.

5.3.2 Wnioskowanie integracyjne

W bazie wiedzy zbudowanej według metody SIM dochodzi do przepływu wniosków pomiędzy kontekstami połączonymi relacją agregacji. Informacje umieszczone w jednym kontekście mogą tym samym wpływać na informacje umieszczone gdzie indziej, przy czym stopień tego wpływu zależy od siły powiązań pomiędzy terminami zdefiniowanymi na poszczególnych poziomach hierarchii SIM. W skrajnym przypadku (oczywiście nieprzydatnym z praktycznego punktu widzenia) można sobie wyobrazić bazę wiedzy z rozgałęzioną hierarchią agregacji kontekstów, przy czym wszystkie te konteksty mogą być równoważnych (posiadających taką samą lub zbliżoną zawartość) typów. Wówczas informacja wprowadzona do jednego kontekstu będzie musiała być rozpropagowana wraz z wnioskami w całej bazie wiedzy.

Dyskusja przytoczona powyżej daje motywację dla opracowania algorytmu *wnioskowania integracyjnego*. W algorytmie tym zawartość kontekstowej bazy wiedzy jest tłumaczona i przekształcana w pojedynczą zintegrowaną, niezmodularyzowaną bazę wyrażoną w tradycyjnej logice opisowej. Wnioskowanie przeprowadzane jest właśnie na tej zintegrowanej bazie.

Takie podejście do wnioskowania nie pozwala wykorzystać dekompozycji wprowadzanej przez konteksty do usprawnienia procesu wyciągania wniosków. Jest jednak

bardzo przydatne ze względów poznawczych, gdyż umożliwia przeniesienie rezultatów dotyczących rozstrzygalności i złożoności wnioskowania w niezmodularyzowanych bazach wiedzy na bazy zbudowane według modelu SIM. Od strony praktycznej z kolei pozwala na zbudowanie silnika wnioskującego wykorzystującego istniejące już narzędzia, takie jak np. Pellet, Racer Pro czy FaCT++ (patrz punkt 2.2.3).

5.3.2.1 Algorytm wnioskowania integracyjnego

Algorytm wnioskowania integracyjnego wykorzystuje metody wnioskowania opracowane dla poszczególnych logik opisowych (na przykład algorytm *tableau*). Zasadniczą nową częścią algorytmu jest mechanizm tłumaczący, który, otrzymując na wejściu kontekstową bazę wiedzy \mathcal{K} , tłumaczy ją na niezmodularyzowaną („płaską”) bazę zintegrowaną („globalną”) KB .

Przy tłumaczeniu bazy wiedzy należy zwrócić szczególną uwagę na dwa zagadnienia. Zagadnienie pierwsze, to koncepty i role, które w bazie \mathcal{K} mają tę samą nazwę, ale zastosowane w dwóch różnych wystąpieniach kontekstów umieszczonych równolegle w hierarchii agregacji oznaczają w rzeczywistości co innego. Dobrą ilustracją tego problemu stanowi umieszczony w punkcie 6.2.1 przykład 6.1. W przykładzie tym osobnik *jan_kowalski* należy jednocześnie do konceptu *Lekarz* oraz do konceptu \neg *Lekarz*, oczywiście w dwóch różnych wystąpieniach kontekstów. Aby zasymulować ten efekt w niezmodularyzowanej bazie wiedzy, zastosowano specjalny przedrostek odróżniający od siebie koncepty w poszczególnych wystąpieniach kontekstów. Mechanizm nadawania przedrostków musi jednak działać tak, aby odróżnienie dotyczyło tylko tych konceptów, które zostały nowo wprowadzone przez typ danego wystąpienia (np. koncept *Umie_reanimować* powinien zachować to samo znaczenie we wszystkich wystąpieniach kontekstów).

Drugie zagadnienie związane z tłumaczeniem wiąże się z możliwością ograniczenia dziedziny w kontekście agregowanym. Aby zamodelować ten efekt w bazie KB do przetłumaczonej bazy wprowadzana jest rodzina *konceptów dziedziny* ($i:\top$) oznaczających dziedziny poszczególnych wystąpień kontekstów. Z kolei każde zdanie wypowiedziane na temat konceptów wykorzystywanych w danym wystąpieniu kontekstu jest przekształcane tak, aby ograniczyć zasięg jego oddziaływania jedynie do dziedziny danego wystąpienia kontekstu.

Przekształcenie bazy wiedzy do postaci zintegrowanej działa zgodnie z koncepcjami przedstawionymi powyżej. Prezentowana tu metoda wykorzystuje pewne konstrukcje z [BS2003] i adaptuje je do specyfiki modelu SIM. Bez zmniejszenia ogólności można przyjąć, że wejściowa baza \mathcal{K} ma strukturę zawierającą wystąpienia kontekstów $\mathbf{A} = \{A_1, \dots, A_n\}$. Na początek należy rozważyć wariant tej struktury bez wieloagregacji. Pozwala to na zdefiniowanie dla każdego niepodstawowego wystąpienia kontekstu A_i , $i \in [1, n]$, jego bezpośredniego agregatu $\pi(i) = j \leftrightarrow A_i \ll \bullet A_j; j \in [1, n]$.

Podstawową rolę w proponowanym algorytmie odgrywa przekształcenie terminów i wyrażeń logiki opisowej występujących w aksjomatach i asercjach tworzących zawartość kontekstów. Definicja przekształcenia $\#(A_i, E)$ zamieniającego wyrażenie E wykorzystane w wystąpieniu A_i jest rekurencyjna. Poniżej została ona podana w wersji dla logiki *SHIQ* (A to koncept atomowy, R to rola, C i D to dowolne koncepty złożone):

$$\begin{aligned}
 \#(A_i, \top) &= i:\top \\
 \#(A_i, \perp) &= \perp \\
 \#(A_i, A) &= i:A && \text{dla } A \in \mathcal{S}(f_{inst}(A_i)) \setminus \mathcal{S}(f_{inst}(A_{\pi(i)})) \\
 &= i:\top \sqcap \#(A_{\pi(i)}, A) && \text{dla } A \in \mathcal{S}(f_{inst}(A_{\pi(i)}))
 \end{aligned}$$

$$\begin{aligned}
\#(A_i, R) &= i:R && \text{dla } R \in \mathcal{S}(f_{inst}(A_i)) \setminus \mathcal{S}(f_{inst}(A_{\pi(i)})) \\
&= \#(A_{\pi(i)}, R) && \text{dla } R \in \mathcal{S}(f_{inst}(A_{\pi(i)})) \\
\#(A_i, R^-) &= \#(A_{\pi(i)}, R)^- \\
\#(A_i, \neg C) &= i:\top \sqcap \neg \#(A_i, C) \\
\#(A_i, C \sqcap D) &= i:\top \sqcap (\#(A_i, C) \sqcap \#(A_i, D)) \\
\#(A_i, C \sqcup D) &= i:\top \sqcap (\#(A_i, C) \sqcup \#(A_i, D)) \\
\#(A_i, \exists R.C) &= i:\top \sqcap \exists \#(A_i, R).\#(A_i, C) \\
\#(A_i, \forall R.C) &= i:\top \sqcap \forall \#(A_i, R).\#(A_i, C) \\
\#(A_i, \geq n R.C) &= i:\top \sqcap \geq n \#(A_i, R).\#(A_i, C) \\
\#(A_i, \leq n R.C) &= i:\top \sqcap \leq n \#(A_i, R).\#(A_i, C)
\end{aligned}$$

Zasadą ogólną jest, że każdy koncept atomowy wprowadzany w ramach wystąpienia kontekstu A_i (tj. nie będący częścią sygnatury typu jego bezpośredniego agregatu) jest rozszerzany o prefiks $i:A$. Z kolei każde wyrażenie conceptowe użyte w wystąpieniu A_i jest przecinane ze specjalnym conceptem $i:\top$ reprezentującym dziedzinę A_i . W ogólności dla konstruktora conceptu złożonego $\rho(E_1, E_2, \dots, E_k)$ przyjmującego k wyrażeń — conceptów i ról — wynikiem przekształcenia jest $i:\top \sqcap \rho(\#(A_i, E_1), \#(A_i, E_2), \dots, \#(A_i, E_k))$, przy czym wyrażenia postaci $\geq n R.C$ i $\leq n R.C$ traktowane są w takim przypadku jako rodziny konstruktorów.

Translacja terminologicznej części bazy \mathcal{K} do bazy KB przebiega z użyciem zdefiniowanego przekształcenia $\#$. Dla każdego wystąpienia A_i wprowadzany jest zbiór terminologii $T(A_i)$, przy czym $T \in T(A_i)$ wtóg $T = f_{inst}(A_i)$ lub $f_{inst}(A_i) \sqsubseteq T$ oraz $T \neq f_{inst}(A_{\pi(i)})$ i nieprawda, że $f_{inst}(A_{\pi(i)}) \sqsubseteq T$. Innymi słowy $T(A_i)$ jest zbiorem typów kontekstów, które leżą na ścieżce dziedziczenia między typami $f_{inst}(A_i)$ i $f_{inst}(A_{\pi(i)})$ z wyłączeniem typu, którego wystąpieniem jest jego bezpośredni agregat $A_{\pi(i)}$. Przy tych założeniach dla wszystkich wystąpień A_i wykonywane jest tłumaczenie według następujących zasad (dla uproszczenia zakłada się, że zawartość typów kontekstów składa się jedynie z aksjomatów podrzędności $C \sqsubseteq D$):

Algorytm 5.1 (przekształcenie do bazy integracyjnej)

Wejście: Kontekstowa baza \mathcal{K} ; Wyjście: Zintegrowana baza KB

Dla każdego wystąpienia kontekstu A_i z \mathcal{K} wykonaj następujące kroki:

- 1) Dla każdego aksjomatu $C \sqsubseteq D$ zawartego w którejś z terminologii w $T(A_i)$ wstaw do KB aksjomat:
 $\#(A_i, C) \sqsubseteq \#(A_i, D)$.
- 2) Dla każdego conceptu atomowego $A \in \mathcal{S}(f_{inst}(A_i)) \setminus \mathcal{S}(f_{inst}(A_{\pi(i)}))$ wstaw do KB aksjomat:
 $\#(A_i, A) \sqsubseteq i:\top$.
- 3) Dla każdej roli $R \in \mathcal{S}(f_{inst}(A_i)) \setminus \mathcal{S}(f_{inst}(A_{\pi(i)}))$ wstaw do KB aksjomaty:
 $\exists \#(A_i, R).\top \sqsubseteq i:\top$
 $\top \sqsubseteq \forall \#(A_i, R).i:\top$.
- 4) Jeśli $A_i = A_{max}$ to wstaw do KB aksjomat:
 $\top \sqsubseteq i:\top$
w przeciwnym przypadku aksjomat:
 $i:\top \sqsubseteq \pi(i):\top$.
- 5) Jeśli w A_i nie ma żadnej asercji, wstaw do KB aksjomat:
 $i:\top(i : a)$,
gdzie $i:a$ jest specjalną nazwą osobnika nie używaną nigdzie w bazie.

- 6) Dla każdej asercji $C(a)$ zawartej w A_i wstaw do KB asercję:
 $\#(A_i, C)(a)$.
- 7) Dla każdej asercji $R(a, b)$ zawartej w A_i wstaw do KB asercję:
 $\#(A_i, R)(a, b)$.
- 8) Dla każdego osobnika zawartego w $\mathcal{S}(A_i) \setminus \bigcup_{j: \pi(j)=i} \mathcal{S}(A_j)$ wstaw do KB asercję:
 $i:\top(a)$.

■

Tak utworzona baza wiedzy KB może posłużyć do wnioskowania, którego wynik można odnieść do zawartości bazy \mathcal{K} . Mówi o tym poniższe twierdzenie:

Twierdzenie 5.1 (spójność bazy zintegrowanej)

Niech \mathcal{K} będzie kontekstową bazą wiedzy bez wieloagregacji, a KB wygenerowaną na jej podstawie za pomocą algorytmu 5.1 bazą zintegrowaną. Wówczas zachodzi następująca zależność: \mathcal{K} jest spójna (ma model kontekstowy) wtwg KB jest spójna (ma model). ■

Dowód:

Dowód jest przeprowadzony poprzez tłumaczenie modelu bazy \mathcal{K} na model bazy KB i odwrotnie.

(\Rightarrow) Weźmy dowolną interpretację kontekstową $\mathcal{J}_{\mathcal{K}} = \{(A_1, \mathcal{J}(A_1)), (A_2, \mathcal{J}(A_2)), \dots, (A_n, \mathcal{J}(A_n))\}$ będącą modelem \mathcal{K} . Na jej podstawie zbudujemy model \mathcal{J}_{KB} . Model ten budujemy w następujący sposób:

$$\Delta^{\mathcal{J}_{KB}} = \bigcup_{i \in [1..n]} \Delta^{\mathcal{J}(A_i)} \quad (5.1)$$

$$(i:\top)^{\mathcal{J}_{KB}} = \Delta^{\mathcal{J}(A_i)} \quad \forall i \in [1, n] \quad (5.2)$$

$$(i:A)^{\mathcal{J}_{KB}} = A^{\mathcal{J}(A_i)} \quad \forall i \in [1, n], \forall A \in \mathcal{S}(f_{inst}(A_i)) \setminus \mathcal{S}(f_{inst}(A_{\pi(i)})) \quad (5.3)$$

$$(i:R)^{\mathcal{J}_{KB}} = R^{\mathcal{J}(A_i)} \quad \forall i \in [1, n], \forall R \in \mathcal{S}(f_{inst}(A_i)) \setminus \mathcal{S}(f_{inst}(A_{\pi(i)})) \quad (5.4)$$

$$a^{\mathcal{J}_{KB}} = a^{\mathcal{J}(A_i)} \quad \forall i \in [1, n], \forall a \in \mathcal{S}(A_i) \quad (5.5)$$

$$(i:a)^{\mathcal{J}_{KB}} = x^{\mathcal{J}(A_i)} \quad \forall i \in [1, n], \text{ takiego że } A_i \text{ jest puste; } x^{\mathcal{J}(A_i)} \text{ to} \quad (5.6)$$

dowolny element $\Delta^{\mathcal{J}(A_i)}$

Na początek należy zauważyć, że \mathcal{J}_{KB} jest zbudowana poprawnie. Wątpliwości może budzić tu zależność (5.5), ale na mocy definicji 5.9 te same nazwy osobników muszą być interpretowane tak samo przez wszystkie interpretacje lokalne.

Dla tak zbudowanej interpretacji \mathcal{J}_{KB} dla dowolnego wystąpienia kontekstu A_i i dowolnego konceptu (być może złożonego) C wyrażonego za pomocą sygnatury właściwej dla A_i zachodzi następująca zależność:

$$\#(A_i, C)^{\mathcal{J}_{KB}} = C^{\mathcal{J}(A_i)} \quad (5.7)$$

Zależność (5.7) jest oczywiście spełniona dla konceptów atomowych. Jej prawdziwości dla konceptów złożonych dowodzimy poprzez indukcję strukturalną. Na przykład, przy założeniu, że $\#(A_i, C)^{\mathcal{J}_{KB}} = C^{\mathcal{J}(A_i)}$ oraz $\#(A_i, D)^{\mathcal{J}_{KB}} = D^{\mathcal{J}(A_i)}$, pokazujemy, że $\#(A_i, C \sqcap D)^{\mathcal{J}_{KB}} = (i:\top \sqcap (\#(A_i, C) \sqcap \#(A_i, D)))^{\mathcal{J}_{KB}} = (\Delta^{\mathcal{J}(A_i)} \cap C^{\mathcal{J}(A_i)} \cap D^{\mathcal{J}(A_i)}) = (C \sqcap D)^{\mathcal{J}(A_i)}$.

Teraz wystarczy pokazać, że \mathcal{J}_{KB} spełnia wszystkie asercje i aksjomaty wprowadzone przez poszczególne kroki algorytmu 5.1:

Krok 1: Na mocy zależności (5.7) każdy aksjomat $\#(A_i, C) \sqsubseteq \#(A_i, D)$ jest spełniony, gdyż interpretacja \mathcal{J}_i będąca modelem \mathcal{K} musi spełniać aksjomat $C \sqsubseteq D$, czyli $C^{\mathcal{J}(A_i)} \subseteq D^{\mathcal{J}(A_i)}$.

Krok 2: Każdy aksjomat $\#(A_i, A) \sqsubseteq i:\top$ jest spełniony na podstawie zależności (5.2) i (5.3) oraz faktu, że $A^{\mathcal{J}(A_i)} \subseteq \Delta^{\mathcal{J}(A_i)}$.

Krok 3: Analogicznie jak dla kroku 2, na mocy faktu, że $R^{\mathcal{J}(A_i)} \subseteq \Delta^{\mathcal{J}(A_i)} \times \Delta^{\mathcal{J}(A_i)}$.

Krok 4: Aksjomaty $i:\top \sqsubseteq \pi(i):\top$ są spełnione na mocy zależności (5.2) oraz zależności między dziedzinami dla interpretacji lokalnych (punkt 1) definicji 5.9). Na mocy tego samego punktu spełniony jest również aksjomat $\top \sqsubseteq i:\top$ dla $A_i = A_{max}$, gdyż $\Delta^{\mathcal{J}(A_i)} = \bigcup_{j \in [1, n]} \Delta^{\mathcal{J}(A_j)}$.

Krok 5: Asercje $i:\top(i:a)$ są spełnione na mocy zależności (5.6).

Krok 6: Asercje $\#(A_i, C)(a)$ są spełnione na mocy zależności (5.7) oraz (5.5).

Krok 7: Asercje $\#(A_i, R)(a, b)$ są spełnione na mocy zależności (5.4) i (5.5).

Krok 8: Asercje $i:\top(a)$ są spełnione na mocy zależności (5.5), faktu, że te same nazwy osobników muszą być interpretowane tak samo przez wszystkie interpretacje lokalne (punkt 2) definicji 5.9) i faktu, że $a^{\mathcal{J}(A_i)} \subseteq \Delta^{\mathcal{J}(A_i)}$.

(\Leftarrow) Weźmy dowolny model \mathcal{J}_{KB} bazy KB . Na jego podstawie zbudujemy interpretację kontekstową $\mathcal{J}_{\mathcal{K}} = \{(A_1, \mathcal{J}(A_1)), (A_2, \mathcal{J}(A_2)), \dots, (A_n, \mathcal{J}(A_n))\}$ będącą modelem \mathcal{K} . Interpretację tę budujemy następująco:

$$\Delta^{\mathcal{J}(A_i)} = (i:\top)^{\mathcal{J}_{KB}} \quad \forall i \in [1, n] \quad (5.8)$$

$$A^{\mathcal{J}(A_i)} = \#(A_i, A)^{\mathcal{J}_{KB}} \quad \forall i \in [1, n], \forall A \in \mathcal{S}(f_{inst}(A_i)) \quad (5.9)$$

$$R^{\mathcal{J}(A_i)} = \#(A_i, R)^{\mathcal{J}_{KB}} \cap (\Delta^{\mathcal{J}_i} \times \Delta^{\mathcal{J}_i}) \quad \forall i \in [1, n], \forall R \in \mathcal{S}(f_{inst}(A_i)) \quad (5.10)$$

$$a^{\mathcal{J}(A_i)} = a^{\mathcal{J}_{KB}} \quad \forall i \in [1, n], \forall a \in \mathcal{S}(A_i) \quad (5.11)$$

Najpierw należy pokazać, że tak skonstruowana funkcja $\mathcal{J}_{\mathcal{K}}$ jest interpretacją kontekstową. Po pierwsze poszczególne interpretacje lokalne spełniają definicję interpretacji: $\Delta^{\mathcal{J}(A_i)}$ są niepuste (co wynika z tego, że \mathcal{J}_{KB} jest modelem KB , a KB zawiera przynajmniej jedną asercję $i:\top(a)$ — kroki 5 i 8 algorytmu 5.1), $A^{\mathcal{J}(A_i)} \subseteq \Delta^{\mathcal{J}(A_i)}$ (co wynika z zależności (5.9) i definicji przekształcenia $\#$), $R^{\mathcal{J}(A_i)} \subseteq \Delta^{\mathcal{J}(A_i)} \times \Delta^{\mathcal{J}(A_i)}$ (co wynika bezpośrednio z zależności (5.10)) oraz $a^{\mathcal{J}(A_i)} \in \Delta^{\mathcal{J}(A_i)}$ (co wynika z zależności (5.11) oraz faktu, że \mathcal{J}_{KB} musi spełniać zdania wygenerowane w krokach 4 i 8 algorytmu 5.1).

$\mathcal{J}_{\mathcal{K}}$ jest interpretacją kontekstową, gdyż spełnia oba punkty definicji 5.9. Dziedziny interpretacji lokalnych zawierają się w sobie zgodnie z hierarchią agregacji, gdyż ekstensje konceptów $(i:\top)^{\mathcal{J}_{KB}}$ muszą się w sobie zawierać w taki właśnie sposób ze względu na fakt, że \mathcal{J}_{KB} spełnia aksjomaty wygenerowane w kroku 4 algorytmu 5.1. Z kolei tej samej nazwie osobnika przypisywany jest ten sam element wszystkich dziedzin na bezpośrednio na podstawie zależności (5.11).

Kolejnym krokiem dowodu jest pokazanie, że $\mathcal{J}_{\mathcal{K}}$ jest modelem bazy \mathcal{K} . W tym celu posłużymy się jeszcze raz zależnością (5.7), która w tej części dowodu nadal obowiązuje, co można pokazać, używając analogicznych jak poprzednio metod. Wykorzystując tę zależność, można pokazać, że $\mathcal{J}_{\mathcal{K}}$ spełnia wszystkie punkty definicji 5.10 (modelu kontekstowej bazy wiedzy \mathcal{K}):

Punkt 1: Wszystkie asercje zawarte w każdym A_i są spełnione przez $\mathcal{J}(A_i)$. Na mocy zależności (5.7) wystarczy pokazać, że dla każdej asercji unarnej $C(a)$ w KB spełniona jest asercja $\#(A_i, C)(a)$, zaś dla każdej asercji binarnej asercje $\#(A_i, R)(a, b)$, $i:\top(a)$, $i:\top(b)$. W pierwszym przypadku wynika to z postaci kroku 6 algorytmu 5.1, w drugim z postaci kroków 7 i 8 tego algorytmu.

Punkt 2: $\mathcal{J}(A_i)$ spełnia wszystkie aksjomaty zawarte w $T(A_i)$, co wynika z zależności (5.7) oraz z postaci kroku 1 algorytmu 5.1.

Punkt 3: Warunek $\forall A_j \ll A_i C^{\mathcal{J}(A_j)} = C^{\mathcal{J}(A_i)} \cap \Delta^{\mathcal{J}(A_j)}$ jest spełniony, gdyż na mocy zależności (5.7): $C^{\mathcal{J}(A_j)} = \#(A_j, C)^{\mathcal{J}_{KB}} = (i:\top)^{\mathcal{J}_{KB}} \cap \#(A_i, C)^{\mathcal{J}_{KB}} = C^{\mathcal{J}_i} \cap \Delta^{\mathcal{J}_j}$.

Punkt 4: Warunek $\forall A_j \ll A_i R^{\mathcal{J}(A_j)} = R^{\mathcal{J}(A_i)} \cap \Delta^{\mathcal{J}(A_j)} \times \Delta^{\mathcal{J}(A_i)}$ jest spełniony na mocy zależności (5.10).

QED

Twierdzenie 5.1 ustanawia ważny związek pomiędzy modelami bazy zintegrowanej a modelami bazy kontekstowej. Tym samym pokazuje też, jak w ramach wnioskowania kontekstowego wykorzystać procedury wnioskowania dla tradycyjnej logiki opisowej. Ponieważ problem spójności bazy wiedzy jest jednym z najtrudniejszych problemów decyzyjnych w logice opisowej, jego rozwiązanie dostarcza także procedurę rozstrzygnięcia takich kwestii jak podrzędność i spełnialność konceptów oraz przynależność osobnika do konceptu.

Wniosek 5.1

Algorytm wnioskowania integracyjnego z kontekstowej bazy \mathcal{K} bez wieloagregacji wyrażonej w logice *SHIQ*, składający się z dwóch kroków: (1) przekształcenia bazy kontekstowej \mathcal{K} do bazy KB (algorytm 5.1) oraz (2) stwierdzenia spójności bazy KB , jest kończąca się, poprawną i pełną procedurą decyzyjną rozstrzygającą o spójności bazy \mathcal{K} . Uzupełnienie tego algorytmu o krok (0) dodający do bazy wiedzy KB dla zadanego kontekstu $\kappa_i = (f_{inst}(A_i), A_i)$ zdanie $\#(A_i, C \sqcap \neg D)(x)$, zdanie $\#(A_i, C)(x)$, gdzie x jest dowolną niewykorzystaną nazwą osobnika, lub zdanie $\#(A_i, C)(a)$, przekształca algorytm w procedurę rozstrzygającą odpowiednio o podrzędności konceptów C i D w κ_i , spełnialności konceptu C w κ_i lub przynależności osobnika a do konceptu C w κ_i .

Dowód:

Wniosek jest bezpośrednią konsekwencją twierdzenia 5.1. Przekształcenie bazy za pomocą algorytmu 5.1 jest oczywiście wykonalne w skończonym czasie. Wynika to stąd, że liczba wyrażeń, które podlegają konwersji, jest skończona, a algorytm nie zawiera pętli. Natomiast ze względu na fakt odpowiedniości modeli dowiedziony dla twierdzenia 5.1, fakt spójności bądź niespójności bazy wiedzy KB pociąga za sobą spójność lub niespójność \mathcal{K} .

Modyfikacja algorytmu o krok dodający odpowiednie zdanie jest techniką znaną z klasycznej logiki opisowej. Fakt możliwości jej zastosowania wynika z analogicznej w stosunku do klasycznej logiki opisowej postaci definicji 5.11 oraz z występowania zależności (5.7) dla bazy KB .

QED

Procedurę opisaną w niniejszym punkcie można zaadaptować do logik innych niż *SHIQ*. Dla podzbiorów tej logiki konieczna jest adaptacja przekształcenia $\#$ (według wymienionej ogólnej zasady $\#(A_i, \rho(E_1, E_2, \dots, E_k)) = i:\top \sqcap \rho(\#(A_i, E_1), \#(A_i, E_2), \dots, \#(A_i, E_k))$). Ponieważ algorytm 5.1 nie wykorzystuje innych operatorów niż przecięcie

konceptów, niekwalfikowany kwantyfikator egzystencjalny i kwantyfikator ogólny (krok 3), możliwe jest wykorzystanie dowolnej logiki opisowej o ekspresywności większej lub równej \mathcal{AL} . W rzeczywistości możliwe jest także dostosowanie powyższej metody do logiki \mathcal{SHOIQ} o większej ekspresywności (pokrywającej język OWL-DL 1.0). Obecność konceptów wyliczanych zaciera jednak granicę pomiędzy terminologią a opisem świata, co komplikuje zagadnienie projektowania kontekstowej bazy wiedzy.

Opisana procedura jest także podstawą do wyznaczenia złożoności obliczeniowej procesu wnioskowania oraz sugeruje sposób wyliczenia takiej złożoności. Jako rozmiar problemu można przyjąć właśnie liczbę i rozmiar aksjomatów i asercji w bazie wiedzy, ale uwzględniając fakt, że ten sam koncept w różnych wystąpieniach kontekstów oznacza co innego. Przy takich założeniach rozmiar problemu jest proporcjonalny do wielkości bazy KB , co oznacza, że złożoność problemu wnioskowania jest taka sama, jak złożoność wnioskowania z bazy zapisanej w tradycyjnej logice opisowej (proces tłumaczenia nie zmienia klasy złożoności, gdyż może być wykonany w czasie wielomianowym względem rozmiaru problemu, zaś algorytmy wnioskowania dla wszystkich wymienionych logik są bardziej złożone).

5.3.2.2 Rozszerzenia algorytmu wnioskowania integracyjnego

Metoda opisywana w poprzednim punkcie posiada poważne ograniczenie: dotyczy ona bazy bez wieloagregacji. Problem wnioskowania dla bazy z wieloagregacją jest trudniejszy. Omówimy tutaj stosunkowo prosty przypadek, kiedy nie zachodzi ograniczanie dziedzin pomiędzy kontekstami w ramach hierarchii agregacji. Innymi słowy dziedziny wszystkich kontekstów są równe. W takim uproszczonym przypadku odniesienia przedmiotowe konceptów i ról w ramach kontekstów agregujących i agregowanych muszą być równe. Sytuacja, kiedy jedna nazwa konceptu oznacza co innego w dwóch kontekstach, jest możliwa jedynie wówczas, gdy spełnione są trzy warunki: konteksty te występują równolegle w hierarchii agregacji (1), nie agreguje ich żaden kontekst, w którego sygnaturze znajduje się dany koncept (2) oraz nie zachodzi przypadek, że konteksty te agregują inny wspólny kontekst (3). Naturalną strategią wydaje się zatem pozostanie przy wprowadzaniu prefiksów dla terminów w poszczególnych wystąpieniach kontekstów oraz utożsamianie tych terminów, które są obecne we wspólnych kontekstach agregujących lub agregowanych.

Ze względu na równość dziedzin nie jest także potrzebne wykorzystywanie przekształcenia $\#$. W jego miejsce można użyć znacznie prostszego przekształcenia $pre(A_i, E)$, które obdarza przedrostkiem i : każdy koncept atomowy i każdą rolę atomową w wyrażeniu E . Formalnie definiujemy to przekształcenie rekurencyjnie: $pre(A_i, A) = i:A$ dla konceptu atomowego A , $pre(A_i, R) = i:R$ dla atomowej roli R , zaś dla konstruktora $\rho(E_1, E_2, \dots, E_k)$ konceptu lub roli $pre(A_i, \rho(E_1, E_2, \dots, E_k)) = \rho(pre(A_i, E_1), pre(A_i, E_2), \dots, pre(A_i, E_k))$. Dodatkowo przeddefiniowania wymaga pojęcie bezpośredniego agregatu. W tym przypadku mówimy o zbiorze bezpośrednich agregatów $\pi(i), j \in \pi(i) \leftrightarrow A_i \ll \bullet A_j; j \in [1, n]$.

Przy użyciu tak zdefiniowanych pojęć algorytm tworzenia zintegrowanej bazy wiedzy KB działa następująco:

Algorytm 5.2 (rozszerzone przekształcenie do bazy integracyjnej)

Wejście: Kontekstowa baza \mathcal{K} ; Wyjście: Zintegrowana baza KB

Dla każdego wystąpienia kontekstu A_i z \mathcal{K} wykonaj następujące kroki:

- 1) Dla każdego aksjomatu $C \sqsubseteq D$ zawartego w którejś z terminologii w $T(A_i)$ wstaw do KB aksjomat:
 $pre(A_i, C) \sqsubseteq pre(A_i, D)$.

- 2) Dla każdego $A \in \mathcal{S}(f_{inst}(A_i))$, takiego, że istnieje $j \in \pi(i)$ takie, że $A \in \mathcal{S}(f_{inst}(A_j))$ wstaw do KB aksjomaty:
 $pre(A_i, A) \equiv pre(A_j, A)$
- 3) Dla każdej $R \in \mathcal{S}(f_{inst}(A_i))$, takiej, że istnieje $j \in \pi(i)$ takie, że $R \in \mathcal{S}(f_{inst}(A_j))$ wstaw do R-Boxa KB aksjomat:
 $pre(A_i, R) \equiv pre(A_j, R)$
- 4) Dla każdej asercji $C(a)$ zawartej w A_i wstaw do KB asercję:
 $pre(A_i, C(a))$.

■

Dla algorytmu 5.2 można dowieść podobnej zależności, jak w przypadku algorytmu 5.1:

Twierdzenie 5.2 (spójność rozszerzonej bazy zintegrowanej)

Niech \mathcal{K} będzie kontekstową bazą wiedzy bez ograniczania dziedzin, a KB wygenerowaną na jej podstawie za pomocą algorytmu 5.2 bazą zintegrowaną. Wówczas zachodzi następująca zależność: \mathcal{K} jest spójna (tj. ma model kontekstowy) wtwg KB jest spójna (ma model). ■

Dowód tego twierdzenia przebiega analogicznie, jak w przypadku twierdzenia 5.1, jest jednak znacznie prostszy, gdyż nie uwzględnia możliwości ograniczania dziedzin. Oczywiście, podobnie jak poprzednio, twierdzenie daje podstawy do utworzenia odpowiednich procedur decyzyjnych.

Kluczowym elementem opisanej w tym punkcie metody jest zrównywanie odniesień dziedzinowych konceptów i ról o tych samych nazwach występujących w parze kontekstów pozostających w relacji agregacji. Przy założeniu równych dziedzin aksjomaty tworzone w krokach 2 i 3 algorytmu odpowiadają warunkom nałożonym na model kontekstowej bazy wiedzy przez definicję 5.10. W warunkach wieloagregacji warunki te powodują zrównanie ekstensji konceptów i ról występujących w kilku kontekstach agregujących.

Istnieje możliwość dalszej adaptacji tej metody w kierunku zniesienia tego ograniczającego założenia. Metoda ta również polegałaby na zrównaniu ekstensji konceptów i ról występujących w więcej niż jednym kontekście agregującym, ale tylko na obszarze zawężonej dziedziny danego kontekstu A_i . Jeśli chodzi o koncepty, efekt ten można byłoby osiągnąć stosunkowo prosto, zamieniając aksjomat generowany w kroku 2 na $\#(A_i, A) \equiv i:\top \sqcap \#(A_j, A)$. Niestety ekspresywność języka $SHIQ$ nie pozwala na sformułowanie podobnego aksjomatu dla ról. O ile w przypadku braku wieloagregacji problem ten można ominąć poprzez odpowiednie zawężanie ekstensji wyrażeń kontekstowych, w przypadku agregowania pewnego kontekstu przez kilka kontekstów z daną rolą metoda ta nie zda egzaminu. Rozwiązaniem może być zastosowanie mniej popularnych logik pozwalających na użycie odpowiednich konstruktorów dla ról (przecięcie i rola uniwersalna), takich jak \mathcal{ALB} [T2001]. Jednak dla logik nie posiadających takich konstruktorów stworzenie algorytmu integracyjnego pozostaje problemem otwartym.

5.3.2.3 Uwagi implementacyjne

Algorytm wnioskowania integracyjnego może wykorzystywać dowolny z dostępnych systemów wnioskujących (np. Pellet [SPGKK2007] lub FaCT++ [TH2006]). Konieczna jest jedynie odpowiednia implementacja mechanizmu konwersji generującego zintegrowaną bazę wiedzy. Ponadto konwersję można przeprowadzić raz, a na utworzonej bazie przeprowadzać dowolną liczbę wnioskowań.

W przypadku niektórych rodzajów aktualizacji kontekstowej bazy wiedzy możliwe jest uzupełnienie bazy zintegrowanej bez konieczności jej przebudowy. W szczególności (w wersji algorytmu dla bazy bez wieloagregacji):

- **dodanie asercji** $C(a)$ do istniejącego wystąpienia kontekstu A_i musi spowodować dodanie asercji $\#(A_i, C)(a)$ do zintegrowanej bazy KB ,
- **dodanie asercji** $R(a, b)$ do istniejącego wystąpienia kontekstu A_i musi spowodować dodanie zbioru asercji $\{\#(A_i, R)(a, b), \top_i(a), \top_i(b)\}$ do zintegrowanej bazy KB ,
- **dodanie aksjomatu** $C \sqsubseteq D$ do istniejącego typu kontekstu T_j musi spowodować dodanie do zintegrowanej bazy KB aksjomatów $\#(A_i, C) \sqsubseteq \#(A_i, D)$ dla wszystkich wystąpień kontekstów A_i takich, że, $T_j \in T(A_i)$,
- **dodanie wystąpienia kontekstu** A_i agregowanego przez istniejące wystąpienie A_j musi spowodować dodanie do zintegrowanej bazy KB aksjomatu $\top_i \sqsubseteq \top_j$.

Możliwość aktualizacji zintegrowanej bazy wiedzy wraz ze zmianami w oryginalnej bazie \mathcal{K} jest ważna nie tylko ze względu na fakt unikania powtórnego wykonywania mechanizmu konwersji, ale również ze względu na to, że pozwala wykorzystać inkrementacyjne procedury wnioskowania wprowadzane ostatnio do systemów wnioskujących ([HPS2006]). Bardziej skomplikowane operacje aktualizacji struktury bazy kontekstowej (jak np. zmiana struktury typów) mogą jednak w ogólności wymagać ponownej generacji zintegrowanej bazy wiedzy.

Algorytm wnioskowania zintegrowanego niewątpliwie warto wykorzystywać wszędzie tam, gdzie możliwość skorzystania z istniejących mechanizmów wnioskujących stanowi ważne wymaganie. Jednak należy zauważyć, że ten sposób wnioskowania nie pozwala na użycie struktury bazy kontekstowej do obciążenia pewnych ścieżek wnioskowania. W bazie zintegrowanej nie mamy już dostępu do informacji o powiązaniach między wystąpieniami i typami kontekstów, co sprawia, że nie można jej wykorzystać do usprawnienia przetwarzania. Te cechy algorytmu wnioskowania zintegrowanego sprawiają, że w praktyce najlepiej nadaje się on do wnioskowania z baz wiedzy o stosunkowo skomplikowanych powiązaniach ekstensji konceptów pomiędzy kontekstami. Konstrukcja algorytmu pozwala na zadawanie zapytań dotyczących tych powiązań, tj. można tutaj wyrazić zapytania dotyczące stosunku (podrzędności, rozłączności, równoważności) między konceptami należącymi do różnych typów kontekstów. Algorytm dobrze radzi sobie z aktualizacją bazy wiedzy, w tym części terminologicznej, pod warunkiem, że nie następują zbyt poważne zmiany jej struktury.

Opisywany algorytm można dostosować do przeprowadzania pewnych niestandardowych wnioskowań. Odnajdywanie kontekstu dla określonego warunku może zostać przeprowadzone iteracyjnie (np. odnalezienie kontekstów, dla których $C \sqsubseteq D$, polega na iteracyjnym sprawdzeniu, dla których A_i zachodzi $\#(A_i, C) \sqsubseteq \#(A_i, D)$). Proces ten można usprawnić poprzez zachowanie informacji o sygnaturach poszczególnych typów i wystąpień kontekstów. Przeprowadzanie wnioskowania z części bazy jest możliwe poprzez wygenerowanie bazy zintegrowanej z pominięciem pewnych kontekstów. Konceptyjnie dostosowanie algorytmu do przeprowadzenia tego typu wnioskowania jest proste, natomiast wymaga ono generacji kolejnej wersji bazy zintegrowanej (niepełnej; trzeba wygenerować ją albo przed udzieleniem odpowiedzi na zapytanie, albo należy utrzymywać i aktualizować osobną kopię niezawierającą niektórych kontekstów). Natomiast zagadnienie wyrażania jednych terminów innymi jest dość skomplikowane i nie jest obsługiwane powszechnie przez systemy wnioskujące (choć pewną rekompensatą jest tu fakt, że algorytm umożliwia zadawanie zapytań dotyczących zależności między konceptami z różnych kontekstów).

5.3.3 Wnioskowanie strukturalne

Opisany wcześniej algorytm wnioskowania integracyjnego charakteryzował się wieloma ciekawymi właściwościami. Nie wykorzystywał on jednakże struktury bazy wiedzy w trakcie samego właściwego procesu wnioskowania. W efekcie nie wykorzystywał potencjalnych możliwości dostosowania przepływu wniosków do hierarchii kontekstów.

W tym punkcie opisany zostanie algorytm oparty na innej zasadzie działania. Wykorzystuje on specjalnie zaadaptowaną do potrzeb kontekstowych baz wiedzy metodę kartograficzną [W2007] opracowaną w Katedrze Inżynierii Oprogramowania Politechniki Gdańskiej. W metodzie kartograficznej wnioskowanie wykonuje się jeszcze przed otrzymaniem zapytania: pewien zakres wniosków wyciągany jest z bazy wiedzy bezpośrednio po wykonaniu aktualizacji. Taki sposób postępowania daje więcej możliwości użycia hierarchii agregacji do ustalenia przepływu wniosków.

5.3.3.1 Algorytm wnioskowania strukturalnego

Metoda kartograficzna polega na nałożeniu na dziedzinę podziału względem pewnej grupy parami rozłącznych konceptów zwanych *regionami*. Koncepty te pokrywają swoim odniesieniem przedmiotowym całą dziedzinę, zaś interesujące użytkownika koncepty powinny być wyrażalne za pomocą sumy regionów. W oryginalnej wersji metody kartograficznej, przeznaczonej dla logiki \mathcal{ALC} , jako regiony wykorzystywane były koncepty postaci $L_1 \sqcap L_2 \sqcap \dots \sqcap L_n$, gdzie $L_i \in \{E_i, \neg E_i\}$ zaś zbiór $\{E_i\}$ obejmował wszystkie koncepty atomowe oraz wszystkie koncepty postaci $\forall R.C$ i $\exists R.C$ wykorzystane w terminologii i opisie świata. Zależnie od sformułowanych aksjomatów część regionów jest niespełnialna.

Opis najważniejszych elementów metody kartograficznej na potrzeby modelu SIM zostanie przedstawiony w wersji dla logiki \mathcal{SHIQ} . Jest to podyktowane wciąż zwiększającą się ekspresywnością logiki obsługiwanej przez algorytmy związane z tą metodą (obecna \mathcal{ALCHIQ}), a także zachowaniem spójności opisu z algorytmem wnioskowania integracyjnego.

Wygodnie jest przyjąć, że wykorzystywane będą wyłącznie następujące konstruktory logiki \mathcal{SHIQ} : $\neg C$, $C \sqcap D$, $C \sqcup D$, $\geq n R.C$. Pozostałe konstruktory sprowadza się do nich poprzez następujące przekształcenia:

$$\begin{aligned} \leq n R.C &\Rightarrow \neg \geq (n + 1) R.C \\ \exists R.C &\Rightarrow \geq 1 R.C \\ \forall R.C &\Rightarrow \neg \geq 1 R.\neg C \end{aligned}$$

W oryginalnej metodzie kartograficznej *dekompozycją D* terminologii T nazwano zbiór parami rozłącznych konceptów wyrażonych w $\mathcal{L}_C(\text{Sig}(T))$ pokrywających całą dziedzinę. Dekompozycja zbudowana na bazie $B = \{E_k\}_{k \in [1, m]}$ to zbiór konceptów postaci $L_1 \sqcap L_2 \sqcap \dots \sqcap L_m$, gdzie $L_j \in \{E_k, \neg E_k\}$. Funkcję odwzorowującą terminologię na skończony zbiór konceptów $\{E_k\}$ nazywamy funkcją ustalającą bazę. W ramach metody kartograficznej najczęściej wykorzystywano funkcję f_b , zdefiniowaną następująco (w wersji dla \mathcal{SHIQ}):

$$\begin{aligned} f_b(A) &= \{A\} && \text{gdzie } A \text{ jest konceptem atomowym} \\ f_b(\neg C) &= f_b(C) \\ f_b(C \sqcap D) &= f_b(C) \cup f_b(D) \\ f_b(\geq n R.C) &= \{\geq n R.C\} \cup f_b(C) \\ f_b(C \sqsubseteq D) &= f_b(C) \cup f_b(D) \end{aligned}$$

Ekspresywnością dekompozycji, oznaczaną $\text{Ex}(\mathbf{D})$, nazywa się podzbiór konceptów $\mathcal{L}_C(\text{Sig}(T))$, które można wyrazić za pomocą sumy konceptów należących do dekompozycji.

Funkcja wyznaczająca tę sumę dla dekompozycji \mathbf{D} nazywa się *funkcją dekomponującą* i jest symbolicznie oznaczana $f_{\mathbf{D}}: Ex(\mathbf{D}) \rightarrow \mathcal{P}(\mathbf{D})$. W przypadku dekompozycji zbudowanej na bazie ustalonej za pomocą funkcji f_b ekspresywność dekompozycji obejmuje wszystkie koncepty złożone możliwe do zbudowania za pomocą przecięcia, sumy i negacji z konceptów atomowych i konceptów postaci $\geq n$ R.C wykorzystanych w terminologii T (po ograniczeniu listy konstruktorów $SHIQ$).

Mapą M dla dekompozycji \mathbf{D} terminologii T nazywa się największy podzbiór dekompozycji taki, że zawarte w nim koncepty są spełnialne, tj. $M = \{C \mid C \in \mathbf{D} \wedge T \not\models C \equiv \perp\}$. Za pomocą mapy można orzekać o spełnialności dowolnego konceptu z ekspresywności dekompozycji. Koncept C jest spełnialny wtwg $f_{\mathbf{D}}(C) \cap M \neq \emptyset$.

Rozmieszczeniem r dla dekompozycji \mathbf{D} opisu świata A nazywa się funkcję przyporządkowującą każdemu osobnikowi z $Sig_T(A)$ zbiór regionów, do których może on należeć. Warunkiem jest tutaj, aby dla każdego $C \in r(a)$ istniała pewna interpretacja \mathcal{I} spełniająca $T \cup A$, taka że $\mathcal{I} \models C(a)$. Wszystkie dopuszczalne rozmieszczenia wyznaczają zatem zbiór takich regionów, do których ekstensji możemy zaliczyć danego osobnika, nie rozspójniając bazy wiedzy.

Metoda kartograficzna dostarcza procedur generowania mapy i rozmieszczeń dla zadanych terminologii i opisów świata. Procedury te będą oznaczane odpowiednio jako $\Gamma_M(T)$ oraz $\Gamma_r(A, M)$.

Wykorzystanie metody kartograficznej do wnioskowania z kontekstowych baz wiedzy wymaga jej adaptacji do specyfiki modelu SIM. Choć pojęcia dekompozycji i jej ekspresywności przenoszą się naturalnie do bazy kontekstowej, kluczowe dla metody pojęcia mapy i rozmieszczenia muszą być rozszerzone tak, by odzwierciedlać definicję wynikania kontekstowego dla bazy \mathcal{K} . *Mapą kontekstu* $M(\kappa_i)$ dla dekompozycji \mathbf{D} jest zbiór konceptów $M = \{C \mid C \in \mathbf{D} \wedge \mathcal{K} \not\models_{\kappa_i} C \equiv \perp\}$. *Rozmieszczenie dla kontekstu* $r(\kappa_i)$ i dekompozycji \mathbf{D} to funkcja odwzorowująca zbiór $S_A(\kappa_i)$ nazw osobników w sygnaturze kontekstu κ_i na regiony $\{C \mid C \in \mathbf{D} \wedge \mathcal{K} \not\models_{\kappa_i} \neg C(a)\}$.

Wyznaczenie mapy $M(\kappa_i)$ oraz rozmieszczenia dla kontekstu $r(\kappa_i)$ pozwala na łatwe przeprowadzenie procedury decyzyjnej dla większości standardowych problemów wnioskowania kontekstowego dla konceptów w ramach ekspresywności danej dekompozycji. Na przykład odpowiedź na pytanie o spełnialność konceptu C w danym kontekście κ_i sprowadza się do sprawdzenia, czy $f_{\mathbf{D}}(C) \cap M(\kappa_i) \neq \emptyset$, zaś zbadanie podrzędności konceptu C w stosunku do konceptu D w danym kontekście κ_i jest równoznaczne ze sprawdzeniem, czy $f_{\mathbf{D}}(C) \cap (\mathbf{D} - f_{\mathbf{D}}(D)) \cap M(\kappa_i) \neq \emptyset$ (przy założeniu spójności bazy wiedzy).

Metoda kartograficzna nie daje jednak gotowych procedur generacji mapy i rozmieszczenia kontekstowego. Pierwszą trudnością, jaką należy pokonać, aby utworzyć te procedury, jest kwestia wykorzystanej dekompozycji. Dekompozycje dla poszczególnych kontekstów będą różnić się, gdyż wykorzystują one różne terminologie, w których użyto różnych konceptów atomowych i konceptów postaci $\geq n$ R.C. Jednakże dla konkretnego kontekstu κ_i możliwe jest wykorzystanie funkcji $f_b(T(\kappa_i))$ do wyznaczenia właściwej dla tego kontekstu bazy B_i dekompozycji \mathbf{D}_i . Co więcej, zastosowanie tej funkcji, ze względu na właściwości bazy kontekstowej, gwarantuje, że dla dwóch kontekstów κ_i i κ_j takich, że $\kappa_i \ll \kappa_j$, zachodzi $B_j \subseteq B_i$. Oznacza to, że dekompozycja \mathbf{D}_i jest „dokładniejsza” niż dekompozycja \mathbf{D}_j , tj. każdy koncept z \mathbf{D}_j można wyrazić jako sumę konceptów z \mathbf{D}_i . Odpowiedniość tę można wyrazić funkcją $f_{\mathbf{D}_i \rightarrow \mathbf{D}_j}: \mathbf{D}_i \rightarrow \mathbf{D}_j$, a także przenieść ją na mapy $f_{M(\kappa_i) \rightarrow M(\kappa_j)} = f_{\mathbf{D}_i \rightarrow \mathbf{D}_j} \cap (M(\kappa_i) \times M(\kappa_j))$. Istnienie takiej funkcji oznacza, że każdemu regionowi C mapy kontekstu agregowanego

odpowiada jeden region $D = f_{M(\kappa_i) \rightarrow M(\kappa_j)}(C)$ mapy kontekstu agregującego, natomiast każdemu regionowi C mapy kontekstu agregującego odpowiada zbiór (być może pusty) $f_{M(\kappa_i) \rightarrow M(\kappa_j)}^-(C)$. Dla uproszczenia w dalszej części wywodu relacja $f_{M(\kappa_i) \rightarrow M(\kappa_j)}^-$ oznaczana będzie $f_{M(\kappa_j) \rightarrow M(\kappa_i)}$ i obie konstrukcje traktowane będą jak funkcje odwzorowujące zbiór regionów na zbiór regionów.

Druga trudność polega na konieczności odpowiedniej obsługi przepływu wniosków pomiędzy kontekstami według hierarchii agregacji wystąpień i dziedziczenia typów. Zgodnie z definicją wynikania kontekstowego rozmieszczenia dla poszczególnych kontekstów muszą uwzględniać informacje zawarte w innych kontekstach. Zaimplementowane mechanizmy przepływu wniosków muszą uwzględniać różnice między zastosowanymi dekompozycjami.

Problem utworzenia skutecznej adaptacji metody kartograficznej dla kontekstowej bazy SIM polega zatem na opracowaniu procedur generacji map kontekstów $\Gamma_M(\kappa_i)$ i rozmieszczeń dla kontekstów $\Gamma_r(\kappa_i)$ dla kontekstowej bazy wiedzy \mathcal{K} , które uwzględniałyby wymienione wyżej problemy. Odpowiedni algorytm podany jest poniżej.

Algorytm 5.3 (utworzenie mapy i rozmieszczenia dla kontekstu)

Wejście: Kontekstowa baza \mathcal{K} , zadany kontekst κ_k ;

Wyjście: Mapa M kontekstu κ_k i rozmieszczenie r dla tego kontekstu.

Wykonaj następujące kroki:

- 1) Dla każdego kontekstu κ_i z \mathcal{K} wyznacz mapę $M_i := \Gamma_M(T(\kappa_i))$ dla dekompozycji \mathbf{D}_i zbudowanej na bazie $f_b(T(\kappa_i))$.
- 2) Dla każdego kontekstu κ_i z \mathcal{K} podstaw $S_{Ri} := \emptyset$; $S_{Ci} := \emptyset$.
- 3) Powtarzaj kroki:
 - a) $S_{Ri}^- := S_{Ri}$.
 - b) $S_{Ri} := S_{Ri}^- \cup \bigcup_{j: A_j \ll A_i \vee A_i \ll A_j} \{R(a, b) \mid R(a, b) \in (f_{cont}(A_j) \cup S_{Rj}^-) \wedge a, b \in \mathcal{S}(A_i) \wedge R \in \mathcal{S}(f_{inst}(A_i))\}$
 dopóki $\exists i (S_{Ri}^- \neq S_{Ri})$.
- 4) Powtarzaj kroki:
 - a) $S_{Ci}^- := S_{Ci}$
 - b) $S_i := f_{cont}(A_i) \cup S_{Ri} \cup S_{Ci}^-$
 - c) $r_i := \Gamma_r(S_i)$
 - d) $S_{Ci} := \bigcup_{j: A_j \ll A_i \vee A_i \ll A_j} \{C(a) \mid a \in \mathcal{S}(A_i) \wedge a \in \mathcal{S}(A_j) \wedge C = \sqcup f_{M_j \rightarrow M_i}(r_j(a))\}$
 dopóki $\exists i (S_{Ci}^- \neq S_{Ci})$.
- 5) Jeśli $\forall i, a (r_i(a) \neq \emptyset)$ (to znaczy, gdy baza jest spójna):
 $M := M_k$; $r := r_k$
 w przeciwnym przypadku (to znaczy, gdy baza jest niespójna):
 $M := \emptyset$; $r := \emptyset$

■

Działanie algorytmu polega na generowaniu dodatkowych asercji, które reprezentują ważne wnioski z innych kontekstów połączonych z zadaniem kontekstem relacją agregacji. Dodatkowe asercje ze zbiorów S_{Ri} informują o wystąpieniach ról, które muszą być uwzględnione podczas wnioskowania. Asercje ze zbiorów R_{Ci} stanowią tłumaczenie

(dokonane za pomocą metody kartograficznej) wniosków z kontekstów agregowanych i agregujących dotyczących osobników w zadanym kontekście. Tłumaczenie odbywa się za pomocą wcześniej zdefiniowanych funkcji $f_{M_j \rightarrow M_i}$ wyrażających odpowiedniość między mapami. Warto zwrócić uwagę na fakt, że tłumaczone są nie poszczególne asercje, ale również wszystkie wnioski (całe rozmieszczenie) dotyczące odpowiednich osobników. Zasadnicze części algorytmu działają na zasadzie iteracyjnej; algorytm kończy pracę, gdy w kolejnym kroku nie uda się wyciągnąć dodatkowych wniosków (tj. zbiory dodatkowych asercji z wcześniejszej iteracji — $S_{R_i}^-$, $S_{C_i}^-$ — są takie same, jak otrzymane w bieżącej).

Twierdzenie 5.3 (poprawność algorytmu tworzenia mapy i rozmieszczenia)

Algorytm 5.3 jest poprawny, tj. kończy swoją pracę, a w wyniku zwraca mapę kontekstu i rozmieszczenie dla kontekstu. ■

Dowód:

Aby dowieść, że algorytm 5.3 kończy swoją pracę, należy pokazać, że zakończą się kroki 3 i 4 algorytmu (dla pozostałych kroków jest to oczywiste). W przypadku kroku 3 jest to stosunkowo proste. Można się tu posłużyć następującym rozumowaniem: liczba wszystkich asercji binarnych zawartych w bazie wiedzy jest skończona. Każde przejście kroku 3, które prowadzi do kolejnej iteracji, musi dodać jedną z takich asercji do któregoś ze zbiorów S_{R_i} . Proces musi ulec zakończeniu, najpóźniej wtedy, kiedy wszystkie zbiory S_{R_i} będą zawierały wszystkie asercje binarne zdefiniowane gdziekolwiek w bazie.

Aby udowodnić, że krok 4 zakończy się, trzeba z kolei zauważyć, że rozmieszczenia r_i w każdym kolejnym kroku muszą być mniejsze bądź równe rozmieszczeniom z kroku poprzedniego. Z kolei maksymalna liczba regionów mapy jest skończona i ograniczona z góry rozmiarem odpowiednich dekompozycji, zaś w sytuacji, gdy wszystkie rozmieszczenia są równe rozmieszczeniom z kroku poprzedniego, algorytm kończy pracę.

Aby dowieść, że funkcja generowana przez algorytm jest rozmieszczeniem dla kontekstu, należy pokazać, że w każdym momencie pracy algorytmu dla każdego osobnika, gdy koncept C z mapy M_i nie należy do $r_i(a)$, to z \mathcal{K} wynika, że a nie należy do (ekstensji) C (tj. $\forall a \in \mathcal{S}(A_i) (\forall C \in M_i (C \notin r_i(a) \rightarrow \mathcal{K} \models_{\kappa_i} \neg C(a)))$).

Spełnienia tej zależności można dowieść przez indukcję względem kolejnych iteracji wykonania kroku 4. Należy w tym celu pokazać, że rozszerzony opis świata S_i wynika z \mathcal{K} w kontekście κ_i . Inicjalnie opis ten składa się asercji zawartych w A_i , które oczywiście wynikają z \mathcal{K} w kontekście κ_i , co wynika bezpośrednio z definicji modelu (definicja 5.10). Dodatkowo S_i zawiera asercje postaci $R(a, b)$ pochodzące z wystąpień kontekstu agregowanych przez A_i i agregujących A_i . Wszystkie te asercje muszą również wynikać z \mathcal{K} w kontekście κ_i na mocy punktu 4) definicji 5.10 (czego łatwo dowieść z kolei przez indukcję względem kolejnych iteracji kroku 3).

Skoro w pierwszej iteracji $\mathcal{K} \models_{\kappa_i} S_i$ oraz oczywiście $\mathcal{K} \models_{\kappa_i} T(\kappa_i)$, to rozmieszczenie r_i wyklucza dla osobnika a wyłącznie koncepty C , dla których $\mathcal{K} \models_{\kappa_i} \neg C(a)$ (co z kolei wynika z tego, że $r_i(a) \notin C \rightarrow T(\kappa_i) \cup S_i \models \neg C(a) \rightarrow \mathcal{K} \models_{\kappa_i} \neg C(a)$).

Po dokonaniu tego spostrzeżenia łatwo stwierdzić, że w każdej kolejnej iteracji kroku 4 do opisu S_i dodawane są wyłącznie zdania, które wynikają z \mathcal{K} w kontekście κ_i . Zdania dodawane są w kroku 4d i są one postaci $\sqcup f_{M_j \rightarrow M_i}(r_j(a))(a)$. Wymuszają one przynależność a do (ekstensji) jednego z regionów odpowiadających regionom niewykluczonym przez rozmieszczenia dla kontekstów agregujących A_i i agregowanych przez A_i . Oczywiście gdyby osobnik a należał do jednego z pozostałych regionów (tj. gdyby $\mathcal{K} \models_{\kappa_i} \neg \sqcup f_{M_j \rightarrow M_i}(r_j(a))(a)$ dla pewnego wystąpienia kontekstu A_j znajdującego się z A_i w relacji agregacji), to zaszłoby

rozspójnienie bazy wiedzy, gdyż osobnik a nie mógłby należeć do ekstensji tych samych konceptów w pewnej parze kontekstów wzajemnie się agregujących (spowodowałoby to naruszenie punktu 3) definicji 5.10 metody kontekstowej).

QED

Natychmiastowy wniosek z twierdzenia 5.3 mówi, że możemy wykorzystać algorytm 5.3 do rozstrzygnięcia wszystkich standardowych problemów wnioskowania terminologicznego w bazie kontekstowej, a także problemu przynależności osobnika do konceptu w kontekście. Niewielkie modyfikacje algorytmu i rozszerzenia pozwalają także na rozstrzygnięcie pozostałych problemów wnioskowania z opisu świata.

5.3.3.2 Uwagi implementacyjne

Przy implementacji algorytmu strukturalnego jest bardzo ważne, aby zwrócić uwagę na kilka zagadnień, które mają duży wpływ na jego wydajność. Przede wszystkim rzeczywisty algorytm wyznaczania rozmieszczenia działa inkrementacyjnie, dzięki czemu istnieje możliwość szybkiego wykorzystania wcześniej wyznaczonych wniosków. Algorytm 5.3 również można zaadaptować do działania inkrementacyjnego.

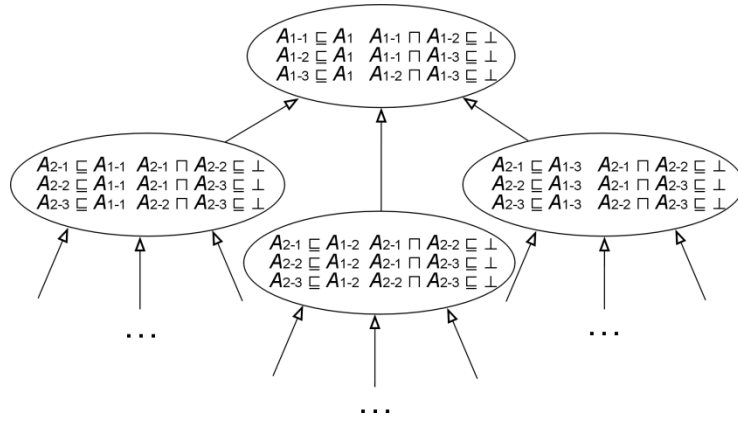
Ilość wniosków zawartych w mapie i rozmieszczeniu jest jednak na tyle duża, że do efektywnego ich przechowywania najlepiej wykorzystać bazę danych. System KASEA, implementujący opisaną metodę w praktyce, używa do tego systemu Oracle 9i. Wykorzystanie bazy danych, oprócz naturalnych korzyści związanych z usprawnieniem obsługi dużych wolumenów danych, pozwala na proste wprowadzenie dodatkowych optymalizacji, takich jak podział osobników na klasy równoważności o takim samym rozmieszczeniu (co pozwala na uniknięcie redundancji przy przechowywaniu informacji o rozmieszczeniu).

Choć, jak należy jeszcze raz podkreślić, mechanizmy oparte na kartografii wiedzy działają zgodnie z filozofią produkcji wniosków metodą w przód, algorytm wnioskowania strukturalnego został zaprezentowany w formie nieinkrementacyjnej. Uczyniono tak, aby ułatwić dyskusję jego podobieństw i różnic w stosunku do algorytmu integracyjnego.

Działanie algorytmu wnioskowania strukturalnego można podzielić na dwie fazy. Pierwsza faza polega na wnioskowaniu lokalnym w ramach wybranego kontekstu (najlepiej reprezentuje ją tworzenie rozmieszczenia lokalnego w kroku 4c). Faza druga polega na „tłumaczeniu” wniosków wysnutych w trakcie wnioskowania lokalnego do kontekstów „sąsiadujących” pod względem relacji agregacji (kroki 3b i 4d). Wykonywanie tych faz przeplata się wzajemnie, aż do osiągnięcia punktu równowagi.

Na potrzeby obecnych rozważań można przyjąć, że przebieg fazy wnioskowania lokalnego jest podobny, jak w przypadku algorytmu *tableau*. Przy takim założeniu dla porównania obu algorytmów należy oszacować potencjalny rozmiar *tableau* (długość ścieżki), zaś dla algorytmu wnioskowania strukturalnego dodatkowy koszt przeprowadzenia tłumaczenia.

Do dalszych rozważań przyjęto następujący scenariusz. Jako podstawową strukturę rozpatrywanej ontologii kontekstowej przyjęto taksonomię z rozłącznością konceptów na każdym poziomie hierarchii. Przyjęto także, że struktura tej taksonomii jest odzwierciedlana przez hierarchię typów w taki sposób, że w ramach każdego typu definiowanych jest b parami rozłącznych konceptów podrzędnych w stosunku do jednego z konceptów wprowadzonych w ramach typu nadrzędnego. Hierarchia typów konceptów (i tak samo taksonomia) formują zatem drzewo o współczynniku rozgałęziania b . Przykład takiej hierarchii dla $b = 3$ pokazano na rysunku 5.4. Przyjęto, że wysokość tej hierarchii może być zmienna i wynosi d .



Rys. 5.4 Hierarchia typów i taksonomia wykorzystana w dyskusji porównawczej dla współczynnika $b = 3$.

Istnieją dwa zasadnicze czynniki wpływające na czas pracy algorytmu *tableau*. Pierwszy czynnik to długość konceptu zinternalizowanego. Koncept zinternalizowany budowany jest jako przecięcie konceptów postaci $C \sqcap \neg D$ dla każdego aksjomatu $C \sqsubseteq D$ zawartego w ontologii. Specyfika działania algorytmu *tableau* sprawia, że wraz z długością konceptu zinternalizowanego rośnie liczba alternatywnych modeli do rozpatrzenia. W omawianym przykładzie można przyjąć, że długość tego konceptu jest proporcjonalna do liczby aksjomatów.

Drugim czynnikiem wpływającym na czas pracy algorytmu jest liczba branych pod uwagę tzw. *etykiety* (ang. *labels*), związana z liczbą konceptów zawartych w terminologii i opisie świata. Liczba ta ogranicza z góry teoretyczną wielkość drzewa modelu rozpatrywanego w algorytmie *tableau*. W omawianym przykładzie zakłada się, że liczba etykiet jest proporcjonalna do liczby wykorzystywanych konceptów atomowych.

Konwersja przykładowej ontologii do ontologii zintegrowanej utworzy w wyniku $(\sum_{i \in [1, d]} b^{(i-1)})(2b + b(b-1)/2 + 1)$ aksjomatów — pierwszy czynnik to liczba kontekstów, składniki sumy drugiego czynnika to kolejno: b aksjomatów postaci $\#(A_i, A_{X-Y}) \sqsubseteq \#(A_i, A_{C-Z})$, b aksjomatów postaci $\#(A_i, A_{X-Y}) \sqsubseteq \top_i$, $b(b-1)/2$ aksjomatów rozłączności oraz jeden aksjomat postaci $\top_i \sqsubseteq \top_j$ — co po uproszczeniu daje liczbę określoną wzorem (5.12).

$$\frac{(b^2 + 3b + 2)(b^d - 1)}{2(b-1)} \quad (5.12)$$

Algorytm wnioskowania integracyjnego musi brać pod uwagę wszystkie koncepty atomowe zdefiniowane w ontologii, czyli $(\sum_{i \in [1, d]} b^{(i-1)})b = (b^{d+1} - b)/(b-1)$ konceptów.

Jako miarę kosztu przeprowadzenia wnioskowania w algorytmie integracyjnym przyjęto większą z wyznaczonych liczb, oznaczoną jako c_{int} . Algorytmy *tableau* dla logik opisowych są zasadniczo wykładnicze (lub nawet klasy Exp2Time), co sugerowałoby dobór miary jako $a^{c_{int}}$, gdzie a to pewna stała. Jednak wysoki stopień optymalizacji algorytmów *tableau* sprawia, że dla dużej części praktycznych problemów koszt obserwowany empirycznie jest znacznie niższy, stąd oszacowanie kosztu jako:

$$c_{int} = \max \left\{ \frac{(b^2 + 3b + 2)(b^d - 1)}{2(b-1)}, \frac{b^{d+1} - b}{b-1} \right\} \quad (5.13)$$

Aby oszacować koszt wykonania algorytmu wnioskowania strukturalnego, należy wziąć pod uwagę jego obie fazy. Miara kosztu fazy wnioskowania lokalnego zostanie uzależniona od identycznych jak poprzednio wartości. W trakcie fazy wnioskowania

lokalnego brane są pod uwagę jedynie aksjomaty z danego typu kontekstu i jego ścieżek dziedziczenia. Największa liczba branych pod uwagę aksjomatów występować będzie zatem w kontekstach o typach położonych najniżej w hierarchii i wynosić będzie $d(b + b(b - 1)/2)$. Natomiast liczba konceptów atomowych w sygnaturze kontekstu wynosi maksymalnie $d \cdot b$.

Aby oszacować wpływ tłumaczenia przyjęto, że wnioskowanie lokalne musi zostać powtórzone tyle razy, ile razy trzeba dokonać korekty rozszerzonego opisu świata. W tym przypadku liczba powtórzeń może wynosić maksymalnie $2d$, czyli podwojona długość najdłuższej ścieżki kontekstów, którą mogą przepływać wnioski. Z kolei przyjmujemy, że koszt samego tłumaczenia jest proporcjonalny do rozmiaru mapy. Rozmiar mapy w podanym przykładzie jest największy dla kontekstów o typach położonych najniżej w hierarchii i wynosi $d(b + 1)$ (dla taksonomii rozłącznej rozmiar ten jest proporcjonalny do liczby konceptów atomowych [W2007]).

Ostatecznie zatem miarę kosztu wykonania algorytmu wnioskowania strukturalnego możemy wyrazić wzorem:

$$c_{str} = 2d \left(\max \left\{ d \left(b + \frac{b(b-1)}{2} \right), d \cdot b \right\} + d(b+1) \right) \quad (5.14)$$

przy czym człon wyznaczający maksimum to miara kosztu wnioskowania lokalnego, $d(b + 1)$ oszacowanie kosztu tłumaczenia, zaś $2d$ to liczba wykonań obu faz. Po uproszczeniu wzoru (5.14) otrzymujemy następującą formułę:

$$c_{str} = d^2 (b^2 + 3b + 2) \quad (5.15)$$

Porównując wzory (5.13) i (5.15), można stwierdzić, że różnią się one znacząco. Dominującym składnikiem oszacowania kosztu dla algorytmu wnioskowania integracyjnego jest wyrażenie eksponencjalne b^d , zaś dla algorytmu wnioskowania strukturalnego to $d^2 \cdot b^2$. Różnica ta wynika z istotnie odmiennej zasady ich działania. Podczas gdy algorytm wnioskowania integracyjnego sprowadza wszystko do pojedynczej ontologii, w algorytmie wnioskowania strukturalnego próbuje się wykorzystać powiązania między typami kontekstów do jak najmocniejszego ograniczenia rozpatrywanej przestrzeni modeli. Opracowany model obliczeń dość dobrze to odzwierciedla: dla niskich wartości b i d koszty wnioskowania są porównywalne, a nawet, że dodatkowy nakład na tłumaczenie sprawia, że algorytm wnioskowania strukturalnego wypada gorzej. Na przykład dla $b = d = 3$ wyznaczone wartości wynoszą odpowiednio $c_{int} = 130$, zaś $c_{str} = 180$. Natomiast dla wyższych wartości b i d koszt wnioskowania integracyjnego rośnie dużo szybciej niż koszt wnioskowania strukturalnego. Dla $b = d = 5$ (363 konceptów) wartości miar wynoszą $c_{int} = 16401$, zaś $c_{str} = 1050$.

Choć różnica w wyznaczonych miarach jest bardzo duża i pokazuje potencjał metody strukturalnej, należy podkreślić fakt, że struktura ontologii w podanym przykładzie została dobrana w ten sposób, aby uwypuklić zalety wnioskowania strukturalnego. W rzeczywistej ontologii liczba iteracji „tłumaczeń” może być znacznie wyższa, a odstępstwa od taksonomicznej budowy hierarchii konceptów mogą skutkować większymi rozmiarami map ([W2007]). Tym niemniej, w nadziei autora, powyższa dyskusja uzasadnia podgląd, że wykorzystanie struktury kontekstów w trakcie wnioskowania daje potencjalnie duże korzyści, jeśli chodzi o wydajność tego procesu.

Dyskusję cech algorytmu wnioskowania strukturalnego warto zamknąć, wskazując, do których rodzajów niestandardowych wnioskowań można go zaadaptować. Podobnie jak w przypadku algorytmu wnioskowania integracyjnego, odnajdywanie kontekstu dla określonego warunku może zostać przeprowadzone iteracyjnie poprzez analizę map. Tutaj wydajność

tego procesu może nawet wzrosnąć, gdyż należy pamiętać, że algorytm wnioskowania strukturalnego przeprowadza wnioskowanie w przód, co oznacza, że potrzebne wnioski będą zapewne już dostępne. Jeśli chodzi o wnioskowanie z części ontologii kontekstowej, to wydaje się, że efektywna adaptacja algorytmu wnioskowania strukturalnego nie jest możliwa. Specyfika jego działania sprawia, że proces wyznaczania map i rozmieszczeń należałoby powtórzyć, co pociągnęłoby za sobą znaczne nakłady. Natomiast problem tłumaczenia terminów z jednego kontekstu na terminy właściwe dla drugiego jest w algorytmie rozwiązywany za pomocą metody kartograficznej. Odpowiednia adaptacja mechanizmu pozwala na przeprowadzenie tego rodzaju wnioskowania.

5.4 Podsumowanie

W niniejszym rozdziale zawarte zostały podstawowe informacje na temat metody SIM. Opisano strukturę kontekstowej bazy wiedzy, jej semantykę, a także opisano system dowodzenia. Wiele miejsca poświęcono na zaprezentowanie problemów związanych z wnioskowaniem. Z opisu wynika, że zaproponowana struktura pozwala rozszerzyć wachlarz podstawowych problemów wnioskowania, dzięki czemu stworzone zostały warunki pod dalszy rozwój związany z wprowadzeniem algebry i kompleksowego języka zapytań.

Zaprezentowano także dwa przykładowe algorytmy wnioskowania. Nie ma wśród nich wersji algorytmu *tableau* dla metody SIM. Jest to pewnym minusem, uniemożliwia bowiem skonfrontowanie prezentowanej metody z innymi metodami modularyzacji, a także uniemożliwia stwierdzenie, czy metoda łągodzi problem wydajnościowy. Jednakże brak tego algorytmu ma swoje uzasadnienie. Jedną z głównych motywacji, którą kierował się autor metody SIM, było, jak powiedziano we wstępie, zaproponowanie takiej metody, która nadawałaby się do realizacji informatycznych systemów szerokiego użytku. Z tego względu istotne jest, aby cechy bazy wiedzy pod pewnymi względami przypominały cechy relacyjnych baz danych. Autor uznał, że zaprezentowany w tym rozdziale algorytm strukturalny dużo lepiej spełnia to wymaganie, dokonując częściowego wnioskowania już w trakcie inicjowania bazy. Algorytm *tableau* posiada cechy, które sprawiają, że nie nadaje się do długotrwałego przechowywania informacji. Najpoważniejszą jego wadą jest niezdolność do obsługi dużych zbiorów faktów. Systemy wnioskujące oparte na tym algorytmie są najczęściej wykorzystywane w trybie jednorazowej sesji, w czasie której po wczytaniu danej terminologii następuje załadowanie niedużego zbioru asercji opisujących daną sytuację, po czym zadawane są zapytania.

W odniesieniu do algorytmu strukturalnego zastosowanie podziału na konteksty daje, jak pokazano w tym rozdziale, bardzo dobre rezultaty. Przedstawione oszacowanie znajdzie swoje potwierdzenie w rozdziale 7. W tym sensie metoda SIM rozwiązuje problem wydajnościowy bardzo skutecznie.

6 Projektowanie hierarchicznych baz wiedzy

Poprzednie rozdziały opisywały metodę hierarchicznej kontekstualizacji SIM w sposób strukturalno-syntaktyczny. Odniosły się także do problemów wnioskowania. Z przedstawionych argumentów wynika, że nie zwiększa ona złożoności zastosowanych algorytmów, a wręcz przeciwnie, może poprzez zawężenie obszaru działania spowodować skrócenie czasu odnalezienia dowodu. Jednakże głównym celem, który został postawiony przed metodą SIM jest rozwiązanie problemów adekwatności kontekstów oraz projektowania kontekstowego, a więc stworzenie środowiska pozwalającego we właściwy sposób wyrazić kontekstowość tworzonego modelu. Adekwatność zaproponowanego środka wyrazu w stosunku do zgodnego z intuicją rozumienia kontekstowości wiedzy ma ułatwić projektowanie ontologii, czyli złagodzić wymienione we wstępie problemy konstrukcyjny i organizacyjny, oraz dostarczyć nowe sposoby korzystania z nich. Ma także pozwolić tworzyć ontologie, które objęłyby swoim opisem większy zakres przewidywanych przez projektanta sytuacji, większą liczbę światów możliwych. Z kolei umożliwienie nowych sposobów korzystania z baz wiedzy mogłoby w rezultacie doprowadzić do upowszechnienia metod sztucznej inteligencji w dziedzinach, w których dotychczas nie były one stosowane.

Metoda hierarchicznej kontekstualizacji ontologii SIM, a przynajmniej zaprezentowany dotychczas formalny opis metody oraz problemów wnioskowania, nie odnosi się bezpośrednio ani do problemów projektowania, ani do sposobów użytkowania. Tworzy jednak środowisko, w którym twórca systemu uzyskuje nowe, niedostępne w innych metodach środki do wyrażania kontekstowych właściwości opisywanej dziedziny. Środowisko to pozwala na stworzenie struktury kontekstów, jednak dotąd brak było informacji w jaki sposób w ramach tych środków realizować potrzeby semantyczne.

W tym rozdziale uwaga jest skierowana na problemy związane z projektowaniem — jak powinna być ukształtowana struktura bazy, jakie konteksty tworzyć, z jakiej przyczyny, jaki ma być ich cel. Umiejętność udzielenia odpowiedzi na te pytania, czyli umiejętność dokonywania świadomych wyborów, jest istotą profesjonalizmu twórców systemów. Dlatego bardzo istotne jest przeanalizowanie wpływu, jaki zaproponowana metoda kontekstualizacji wywiera na kształt tworzonego zgodnie z jej regułami systemu oraz na proces projektowania tych systemów.

W podrozdziale 6.1 naszkicowano ogólnie cechy semantyczne i użytkowe systemów, które są zbudowane zgodnie z zasadami narzucanymi przez metodę SIM. Podrozdział 6.2 jest poświęcony problemom związanym z projektowaniem hierarchiczno-kontekstowej bazy wiedzy. Ostatni podrozdział, 6.3, zawiera podsumowanie rozdziału.

6.1 Cechy systemu zgodnego z SIM

Zanim zostaną przedstawione zagadnienia związane z projektowaniem kontekstowej bazy wiedzy należy zastanowić się nad tym, w jaki sposób właściwości kontekstowe, które metoda SIM pozwala zawrzeć w bazie wiedzy, wpływają na proces jej projektowania i eksploatację. Podstawą do rozważań będzie przedstawiona w punkcie 4.1.1 metafora kontekstu jako pudełka. Zgodnie z tą metaforą kontekst jest pojemnikiem na zdania, których znaczenie jest opisane za pomocą zbioru *parametrów kontekstowych* i ich wartości. Na potrzeby obecnych rozważań należy tę metaforę zmodyfikować, gdyż na etapie projektowania nie należy wnętrza kontekstu identyfikować jeszcze ze zbiorem zdań. Język kontekstu jest dopiero tworzony, zdania dopiero zaczynają być formułowane. Wnętrze kontekstu powinno być zatem kojarzone ze sposobem konceptualizacji. Parametry kontekstowe decydują o zawartości „pudełka”, jednak nie chodzi o ich wpływ na sposób wartościowania zdań, lecz

na sposób wyodrębniania relacji konceptualnych oraz sposób ich nazywania (czyli o tworzone, a nie z góry dany, język). Ogólnie mówiąc, tworzą one metasemantykę ontologii, to znaczy opisują jej semantykę z punktu widzenia innej warstwy abstrakcji.

Z punktu widzenia projektantów i użytkowników parametry kontekstowe porządkują i opisują bazę wiedzy oraz narzucają ograniczenia, które wyznaczają dopuszczalny kształt jej struktury. Dzięki temu opisany zostaje również oczekiwany sposób eksploatacji bazy w ramach systemu zbudowanego zgodnie z metodą SIM. W punkcie 6.1.1, przedstawiono sposób tworzenia kontekstowego *opisu parametrycznego*, na który składa się zbiór parametrów kontekstowych i jego organizacja. Szczególną uwagę zwrócono na wpływ tego opisu na strukturę kontekstowej bazy wiedzy, a także na sposób, w jaki baza ta odzwierciedla pewne zależności w obrębie opisu parametrycznego.

Zaproponowany sposób tworzenia struktury kontekstowej pozwala dokonać podziału bazy wiedzy na *schemat*, czyli część, która jest wynikiem konceptualizacji i *instancję*, czyli część, której zmiana nie narusza już przyjętego modelu, a tylko wzbogaca go o opis faktów. Podział ten jest zdefiniowany w punkcie 6.1.2. Jest on przydatny między innymi do opisanie zakresu kompetencji osób projektujących system i osób, które go eksploatują. Przyjęto mianowicie, że rolą projektanta jest podjęcie wszystkich tych decyzji, które są związane z konceptualizacją, natomiast rolą użytkownika jest opisywanie zjawisk ze świata w ramach narzuconej mu konceptualizacji. Obie role wymagają zupełnie innego przygotowania i zupełnie innego sposobu widzenia problemów związanych z daną dziedziną. Jest to zresztą podział podobny do tego, jaki obecnie istnieje w inżynierii oprogramowania.

Przyjęcie podziału kontekstowej bazy wiedzy oraz wyznaczenie ról osobom uczestniczącym w jej rozwoju tworzy podstawę do formalnego opisu zjawisk, które zachodzą w okresie eksploatacji systemów kontekstowych. Okres ten można przedstawić jako ciąg odcinków czasowych, charakteryzujących się tym, że w ramach każdego z tych odcinków dany schemat połączony jest z inną instancją. Szkic działania takich systemów zamieszczony jest w punkcie 6.1.3.

Punkt 6.1.4 konfrontuje metodę SIM oraz przyjęty model kontekstowego opisu parametrycznego z metodą walidacji ontologii OntoClean, zaprezentowaną w podrozdziale 3.1. Konfrontacja ta pokazuje, w jaki sposób elementy struktury hierarchii kontekstowej porządkują, a jednocześnie uelastyczniają, zasady przyporządkowywania metawłaściwości definiowanym konceptom.

6.1.1 Struktura bazy wiedzy a parametry kontekstowe

Parametry kontekstowe nie są częścią modelu SIM. Nie są też w żaden sposób sformalizowane. Mogą być w trakcie prac nad projektem bazy wiedzy wyrażane za pomocą języka naturalnego. Jednakże powinny być wystarczająco precyzyjne, ażeby jasno opisać przyczynę i cel istnienia kontekstu zarówno członkom zespołu tworzącego bazę wiedzy, jak i osobom ją rozwijającym oraz z niej korzystającym. W tym sensie należy je postrzegać również jako ograniczenie, które zapobiega tworzeniu niepoprawnych, z punktu widzenia twórców systemu, struktur bazy wiedzy. Nie odnosząc się na razie do znaczenia parametrów kontekstowych w odniesieniu do modelowanej dziedziny, zostanie ono zaprezentowane w podrozdziale 6.2, można opisać pewne ich cechy wynikające z formalizmu metody SIM.

Niech P oznacza skończony zbiór parametrów kontekstowych, które biorą pod uwagę twórcy systemu. Każdy z parametrów posiada jakąś dziedzinę, czyli zakres dopuszczalnych wartości. Dziedzina parametru $p \in P$ będzie oznaczana symbolem V_p . Niech $f_p: \mathbf{T} \rightarrow \mathcal{P}(P)$ będzie funkcją przypisującą każdemu typowi kontekstu z kontekstowej terminologii \mathbf{T} podzbiór parametrów ze zbioru P .

Podstawowa cecha funkcji f_P wynika z właściwości semantycznych struktury kontekstowej zaprojektowanej według metody SIM. Przesuwając się w ramach tej struktury w dół hierarchii dziedziczenia (czyli oddalając się od elementu maksymalnego), konteksty dysponują coraz bardziej precyzyjną konceptualizacją oraz opisują coraz mniejszą dziedzinę. Inaczej mówiąc, rośnie poziom szczegółowości kontekstu i maleje jego dziedzina. Jednak w ramach dziedziny kontekstu dziedziczona jest cała konceptualizacja. Oznacza to, że przeznaczenie kontekstu się nie zmienia, lecz zawęża. Wszelkie ograniczenia muszą więc, idąc w dół hierarchii, pozostać w mocy. Każdy typ kontekstu *dziedziczy* więc wszystkie parametry kontekstu dziedziczonego. Może też dodawać swoje własne. Z tego wynika, że funkcja f_P musi danemu typowi kontekstu przypisywać co najmniej te parametry, które zostały przypisane wszystkim typom kontekstów, od których dany typ kontekstu dziedziczy. Formalnie właściwość ta może być wyrażona w sposób następujący:

$$\forall T_1, T_2 \in \mathbf{T} (T_2 \trianglelefteq T_1 \rightarrow f_P(T_1) \subseteq f_P(T_2)) \quad (6.1)$$

W przypadku wielodziedziczenia dany typ kontekstu dziedziczy parametry od wszystkich swoich przodków, czyli wyrażenie (6.1) można zapisać w sposób, który lepiej to zobrazuje:

$$\forall T \in \mathbf{T} (T \trianglelefteq T_{max} \rightarrow \bigcup_{T_i: T \triangleleft T_i} f_P(T_i) \subseteq f_P(T)) \quad (6.2)$$

Przypisanie typowi kontekstu podzbioru parametrów oznacza, że każde nowe wystąpienie tego typu musi być opisane poprzez narzucenie wartości wszystkim parametrom z tego podzbioru. Czyli istnieje funkcja wartościująca f_V , która każdemu wystąpieniu kontekstu A należącemu do kontekstowego opisu świata \mathbf{A} oraz każdemu parametrowi $p \in f_P(f_{inst}(A))$ przyporządkowuje jakąś wartość z dziedziny tego parametru V_p . Zanim jednak zostaną opisane właściwości tej funkcji, należy zastanowić się nad jej charakterem oraz źródłem dziedzin parametrów i przypisywanych im wartości.

Jak to zostało powiedziane wcześniej, parametry kontekstowe nie są bytem sformalizowanym w ramach metody SIM. Jednakże podstawowe źródła informacji, które mogą stanowić przedmiot zawartego w parametrach opisu, wynikają z tej samej konceptualizacji, którą ma opisywać tworzona baza wiedzy. Wyodrębniając encje oraz badając ich właściwości w ramach opisywanej dziedziny, twórcy ontologii muszą podjąć decyzję, które z nich staną się podstawą ustanowienia relacji konceptualnej $r^n: W \rightarrow \mathcal{P}(\Delta^n)$ (patrz punkt 2.1.4). Ustanowienie takiej relacji jest początkiem procesu formalizacji. Z pewnością nie wszystkie dostrzeżone encje i właściwości nadają się do takiej formalizacji (zamiany na elementy podzbiorów dziedziny w sensie teoriomnogościowym). Nie wszystkie też spośród tych, które się nadają, zostaną sformalizowane, gdyż twórcy ontologii kierują się najczęściej chęcią jej ograniczenia do opisu jedynie tych problemów, które są istotne z punktu widzenia przewidywanego zastosowania. Pierwszym źródłem dla parametrów kontekstowych są więc te encje i właściwości, które zostały dostrzeżone w ramach dziedziny, ale nie są przekształcane w relacje konceptualne. Drugie źródło jest związane z tym, że każda ontologia jedynie w sposób przybliżony opisuje zamierzone modele. Niedoskonałość użytej logiki powoduje, że parametry kontekstowe, wyrażane zupełnie innym językiem (na przykład językiem naturalnym) niż ontologia, mogą zwiększyć precyzję opisu. Istnieją także takie źródła danych dla parametrów kontekstowych, które nie mają nic wspólnego z opisywaną dziedziną i z przeprowadzaną konceptualizacją, ale mogą mieć wpływ na sposób tworzenia kontekstów. Najprostszym przykładem takiej informacji jest różnicowanie uprawnień dostępu do danych zawartych w ontologii.

W świetle powyższych rozważań widać, że opisanie właściwości funkcji wartościującej nie jest łatwe. Przynajmniej przede wszystkim trudno jest opisać dziedzinę danego

parametru. W praktyce projektowej dziedziny parametrów kontekstowych należy rozumieć semantycznie. W wielu przypadkach semantyczne znaczenie pobranej z takiej dziedziny wartości da się wyrazić za pomocą prostego symbolu, takiego jak liczba lub nazwa wzięta ze słownika języka naturalnego. Częściej jednak będą to jakieś wartości złożone, na przykład zakresy czasowe lub nieprecyzyjnie opisane położenie w przestrzeni. Nierzadko zapewne będą to nawet opisy, których przetłumaczenie na jednoznaczny symbol będzie wymagało zdefiniowania skomplikowanej funkcji semiotycznej. Dlatego, aby zachować ogólność rozważań, nie należy przyjmować, że semantycznie rozumiana dziedzina parametru składa się z obiektów, które są łatwo dostępne dla funkcji wartościującej. Niech to będą raczej jakieś składniki dziedziny, których cechy nie są znane. Przynależność tych składników do dziedziny będzie oznaczana symbolem \in , jednak nie należy tego traktować w sposób formalny, gdyż nie znaczy to, że dziedzina musi być zbiorem, a składnik elementem tego zbioru. Jedyną cechą składników, która będzie tutaj ważna, jest istnienie zwrotnej i antysymetrycznej relacji kompatybilności, oznaczanej dalej znakiem \Rightarrow . Wyrażenie $v \Rightarrow v'$ będzie oznaczało, że v jest kompatybilne z v' (ale nie odwrotnie). Relacja ta może praktycznie mieć wiele różnych znaczeń. Na przykład, gdy dany parametr ma przechowywać zakres wieku osób, to kompatybilność $v \Rightarrow v'$ może oznaczać, że $v \subseteq v'$. Jeśli z kolei wartości v i v' są jakimiś bardziej skomplikowanymi strukturami (które mogą być na przykład symbolizowane przez grafy), to kompatybilność może oznaczać fakt, że v' da się przekształcić na v za pomocą danego przekształcenia. Relacja kompatybilności nie powinna być przechodnia. To pozwoli łatwiej opisać proces definiowania węzłów agregacji. W przykładzie z zakresami wieku relacja \subseteq jest przechodnia. Można ten problem ominąć w ten sposób, że kompatybilność $v \Rightarrow v'$ zostanie zdefiniowana przez nałożenie większej liczby warunków, na przykład:

$$(v \Rightarrow v') \rightarrow (v \subseteq v' \wedge v = [x, y] \wedge v' = [x', y'] \wedge (y - x \geq y' - x' - 2)),$$

co oznacza, że różnica między zakresami symbolizowanymi przez v i v' musi przekraczać 2 (lata).

Powyższe ustalenia pozwalają już opisać właściwości funkcji wartościującej f_V . Jak już zostało powiedziane, funkcja ta przyporządkowuje każdemu wystąpieniu kontekstu oraz każdemu parametrowi jakąś wartość z dziedziny tego parametru:

$$\text{Fnc}(f_V) \wedge f_V = \{(A, p, v) \mid A \in \mathbf{A} \wedge p \in f_P(f_{inst}(A)) \wedge v \in V_p\}. \quad (6.3)$$

Pierwszą istotną właściwością funkcji jest to, że każdemu wystąpieniu kontekstu danego typu muszą być przyporządkowane wartości parametrów różniące się od wartości przypisanych innym wystąpieniom tego typu:

$$\forall A_1, A_2, A_1 \neq A_2 ((f_{inst}(A_1) = f_{inst}(A_2)) \rightarrow \exists p \in f_P(f_{inst}(A_1)) (f_V(A_1, p) \neq f_V(A_2, p))). \quad (6.4)$$

Właściwość tę można nazwać *zasadą jednoznaczności opisu parametrycznego*, lub po prostu *zasadą jednoznaczności*. Gdyby ta zasada nie obowiązywała, oznaczałoby to, że dobrany zbiór parametrów nie jest wystarczający do opisanie danego kontekstu, że istnieją jakieś nieujawnione parametry opisujące ten kontekst, a taka możliwość jest w prowadzonych tu rozważaniach wykluczona.

Druga właściwość wynika z zasad zachowania modelu określonych w definicji 5.10, które mówią, że w wystąpieniach kontekstów pozostających względem siebie w relacji agregacji wszystkie role i koncepty, z dokładnością do wspólnej części dziedziny, muszą być interpretowane tak samo. Zasady te w odpowiedni sposób przenoszą się na zbiór parametrów i powodują, że wystąpienie agregowane nie może w ramach dziedzicznych parametrów być opisane w sposób sprzeczny, z opisem wystąpienia agregującego. Innymi słowy, wartości

parametrów wspólnych muszą dla wystąpienia agregowanego być kompatybilne z wartościami odpowiednich parametrów dla wystąpienia agregującego:

$$\forall A_1, A_2 \in \mathbf{A} (A_1 \ll A_2 \rightarrow \forall p \in f_P(f_{inst}(A_2))(f_V(A_1, p) \Rightarrow f_V(A_2, p))). \quad (6.5)$$

Zależność kompatybilności między wartościami przypisanymi danemu parametrowi w ramach wystąpień połączonych relacją agregacji zgadza się z intuicyjnym odczuciem, że wystąpienie agregowane jest uszczegółowieniem wystąpienia agregującego i jego zawężeniem. Jeśli na przykład wystąpienie agregujące opisuje uczelnię, to wystąpienie agregowane może opisywać jeden wydział; jeśli wystąpienie agregujące opisuje jeden rok akademicki, to wystąpienie agregowane może opisywać jeden semestr. Jak widać, projektant określając zasady kompatybilności w ramach dziedziny danego parametru, ogranicza zakres wartości tego parametru na danym poziomie agregacji. Można na przykład powiedzieć, że w ramach dziedziny parametru o nazwie *jednostka organizacyjna* katedra jest kompatybilna z wydziałem, ale nie jest kompatybilna z uczelnią. Wtedy wystąpienia, dla których parametr ten przyjmie wartość opisującą daną katedrę będą mogły być agregowane tylko przez wystąpienia, dla których parametr ten przyjmie wartość opisującą dany wydział. Może być też inaczej: można utworzyć odrębne parametry o nazwach *uczelnia*, *wydział*, *katedra* i przyjąć, że typy wystąpień agregowanych w miarę schodzenia w dół relacji agregacji, mają przyporządkowaną coraz większą liczbę parametrów precyzującą miejsce opisywanego działu w danej hierarchii organizacyjnej. Wtedy, w przypadku niektórych parametrów, przesuwanie się w dół relacji agregacji może nie zmieniać przypisanej im wartości. Relacja kompatybilności musi być zdefiniowana jako równość (czyli $v \Rightarrow v' \leftrightarrow v = v'$).

W szczególnym przypadku, gdy kompatybilność wartości wszystkich dziedzicznych parametrów jest równoważna równości, formuła (6.5) stwarza dodatkowe ograniczenie na układ parametrów kontekstowych. Ograniczenie to polega na tym, że typ wystąpienia A_1 albo może mieć tylko jedno wystąpienie agregowane przez A_2 , albo funkcja f_P musi mu przyporządkować więcej parametrów niż ma ich przyporządkowanych typ wystąpienia A_2 :

$$\forall T \in \mathbf{T} (\exists A_1, A_2 (f_{inst}(A_1) = T \wedge A_1 \ll A_2) \rightarrow (\neg \exists A_3 (f_{inst}(A_3) = T \wedge A_3 \ll A_2) \vee (f_P(f_{inst}(A_2)) \subset f_P(T))) \quad (6.6)$$

Dowód:

Przyjmijmy dowolne, ale ustalone wystąpienia kontekstów $A_1, A_2 \in \mathbf{A}$, takich że $A_1 \ll A_2, f_{inst}(A_1) = T_1$ i $f_{inst}(A_2) = T_2$, gdzie $T_1, T_2 \in \mathbf{T}$, oraz $T_1 \preceq T_2$.

Aby udowodnić formułę (6.6) wystarczy dowieść, że gdy nie jest spełniony następnik implikacji, to nie może też być spełniony jej poprzednik. Dla przyjętych założeń zaprzeczeniem następnika będzie formuła:

$$\exists A_3 (f_{inst}(A_3) = T_1 \wedge A_3 \ll A_2) \wedge f_P(T_2) = f_P(T_1). \quad (6.7)$$

Z powyższej formuły, z założeń, z formuły (6.5) oraz po zamianie kompatybilności na równość mamy:

$$\forall p \in f_P(T_2) (f_V(A_1, p) = f_V(A_2, p) = f_V(A_3, p)). \quad (6.8)$$

Ponieważ, zgodnie z (6.7), $f_P(T_2) = f_P(T_1)$, z powyższej formuły wynika, że

$$\neg \exists p \in f_P(f_{inst}(A_1)) (f_V(A_1, p) \neq f_V(A_3, p)), \quad (6.9)$$

zaprzeczeniu ulega poprzednik implikacji zasady jednoznaczności (formuły (6.4)), a więc typy wystąpień A_1 i A_2 nie mogą być równe. Stąd oraz z (6.7) wynika, że typem A_1 nie może być T_1 , a to zaprzecza poprzednikowi formuły (6.6).

QED

Relacja kompatybilności jest relacją równości głównie wtedy, gdy wartościami parametrów są obiekty typu prostego, na przykład liczby, wartości logiczne, typy wyliczeniowe itp. Jak widać z powyższych rozważań, może to ograniczyć swobodne wprowadzanie nowych wystąpień kontekstów. W podobny sposób przyjęcie dziedziny parametru jako zbioru podzbiorów zbioru skończonego i zdefiniowanie relacji kompatybilności jako relacji zawierania może ograniczyć liczbę dopuszczalnych wystąpień. Jeśli na przykład jedynym parametrem, który może odróżniać wystąpienia danego typu jest parametr, którego wartością może być podzbiór właściwy zbioru 3-elementowego, to maksymalna liczba tych wystąpień będzie równa 6 (łącznie ze zbiorem pustym — 7). Dlatego określając zbiór parametrów kontekstowych, należy wziąć pod uwagę powyższe ograniczenie. Z jednej strony może ono być pożytecznym sposobem kontrolowania liczby wystąpień danego typu, z drugiej strony może doprowadzić do utrudnień w korzystaniu z bazy wiedzy lub, co być może jest gorsze, do dezaktualizacji przyjętego zbioru parametrów.

Opisana wyżej właściwość funkcji wartościującej obrazuje również, jak silny jest związek między przeznaczeniem danego wystąpienia kontekstu, wyrażonym wartościami parametrów kontekstowych, a wyborem jego węzła agregacji. Wystąpienia kontekstu typu T agregowane przez wystąpienie, do którego typu wiedzy od T krótsza ścieżka dziedziczenia, mają więcej cech wspólnych niż te, które są agregowane przez wystąpienie, którego typ jest bardziej oddalony od T wzdłuż ścieżki dziedziczenia. Wynika to stąd, że w tym pierwszym przypadku większa liczba wartości parametrów musi być kompatybilna z wartościami parametrów wspólnych, narzuconymi przez wystąpienie agregujące. To wpływa na opis wystąpienia kontekstu — chcąc opisać przyczynę i cel jego istnienia, należy uwzględnić miejsce, w którym będzie ono agregowane. Wyznaczenie dopuszczalnych miejsc agregacji dla danego typu kontekstu jest więc czynnością projektową.

Jak dotąd kontekstowy opis parametryczny był przedstawiany w sposób jak najbardziej przypominający sytuacje rzeczywiste, czyli sytuacje, gdzie wartości parametrów nie są precyzyjnie zdefiniowane i często mogą być zdaniem sformułowanymi w języku naturalnym. Jedynym wymaganiem nałożonym na te wartości było określenie relacji kompatybilności. W ramach przeprowadzanych dalej rozważań przydatne będzie przyjęcie dodatkowego wymagania, że relacja kompatybilności jest relacją równości. Pozwoli to odkryć pewne niewidoczne dotąd zależności między opisem parametrycznym a strukturą kontekstowej bazy wiedzy. Zdefiniowanie relacji kompatybilności jako relacji równości było już omawiane wyżej. Przyjęcie, że wszystkie parametry mają mieć tak właśnie zdefiniowaną kompatybilność bazuje na założeniu, że parametry posiadające złożoną strukturę wewnętrzną dadzą się rozbić na takie parametry składowe, dla których relacja kompatybilności da się przedstawić jako równość części parametrów składowych. Na przykład jeśli dany parametr wyraża jakiś okres czasu i relacja kompatybilności jest zdefiniowana jako zawieranie się, to powinien on zostać rozbity na parametry symbolizujące podokresy o różnej ziarnistości: *rok szkolny, semestr, miesiąc...*

Przyjęcie silniejszych wymagań pozwala zdefiniować pojęcie *struktury regularnej* kontekstowej bazy wiedzy. Struktura ta jest zbudowana według następujących zasad. Funkcja f_P jest definiowana tak jak wyżej, czyli $f_P: \mathbf{T} \rightarrow \mathcal{P}(P)$, z jedną zmianą polegającą na tym, że w wyrażeniu (6.2) zawieranie niewłaściwe zamienione jest na zawieranie właściwe:

$$\forall T \in \mathbf{T} (T \trianglelefteq T_{max} \rightarrow \bigcup_{T_i: T \trianglelefteq T_i} f_P(T_i) \subset f_P(T)) \quad (6.10)$$

Podstawowa definicja funkcji f_V (formuła (6.3)) pozostaje w mocy. Zmienia się zasada jednoznaczności (formuła (6.4)), jednak aby przedstawić nową jej wersję należy zdefiniować pojęcie rzutu regularnego wystąpienia kontekstu na typ dziedziczony.

Definicja 6.1 (rzut regularny wystąpienia kontekstu)

Rzutem regularnym $A_1[T_1]$ wystąpienia kontekstu A_1 na typ T_1 , taki że $f_{inst}(A_1) \preceq T_1$ lub $f_{inst}(A_1) = T_1$, nazywa się:

- 1) jeżeli $f_{inst}(A_1) = T_1$, to A_1 ,
- 2) jeśli istnieje A_2 , takie że: $f_{inst}(A_2) = T_1 \wedge A_1 \ll A_2$, to A_2 ,
- 3) w pozostałych przypadkach nowe wystąpienie kontekstu A_3 , takie że: $f_{inst}(A_3) = T_1 \wedge f_{cont}(A_3) = \{ \} \wedge \forall p \in f_P(T_1) (f_V(A_3, p) = f_V(A_1, p)) \wedge A_1 \ll A_3 \wedge A_3 \ll \bullet A_{max}$.

■

Zasada jednoznaczności (formuła (6.4)) przyjmuje w ramach struktury regularnej następującą postać:

$$\forall T (\forall A_1, A_2 (((f_{inst}(A_1) = T \vee f_{inst}(A_1) \preceq T) \wedge (f_{inst}(A_2) = T \vee f_{inst}(A_2) \preceq T) \wedge A_1[T] \neq A_2[T]) \rightarrow \exists p \in f_P(T) (f_V(A_1[T], p) \neq f_V(A_2[T], p)))). \quad (6.11)$$

Mówi ona, że nie tylko wystąpienia danego typu T muszą różnić się między sobą chociaż jedną wartością parametru, lecz także rzuty wszystkich wystąpień typów dziedziczących od T na typ T muszą spełniać tę zasadę. Po zmodyfikowaniu nowa postać zasady nazywa się *zasadą jednoznaczności regularnej*.

Formuła (6.5) również ulega zmianie. Zmiana ta polega na tym, że relacja kompatybilności jest wymieniana na równość:

$$\forall A_1, A_2 \in \mathbf{A} (A_1 \ll A_2 \rightarrow \forall p \in f_P(f_{inst}(A_2)) (f_V(A_1, p) = f_V(A_2, p))). \quad (6.12)$$

Struktura regularna ściślej odpowiada kontekstowemu opisowi parametrycznemu. Poza tym silniej zaznacza związek, jaki istnieje między konceptualizacją a miejscami agregacji. Na przykład prawdziwe jest następujące twierdzenie:

Twierdzenie 6.1 (o konieczności ustanowienia agregatu)

Jeśli w ramach danej struktury regularnej dane dwa wystąpienia kontekstów A_1 i A_2 posiadają wspólny typ dziedziczony T_1 , taki że dla wszystkich parametrów przyporządkowanych do typu T_1 wystąpienia A_1 i A_2 mają przyporządkowane równe sobie wartości, i nie istnieje żaden typ T_2 dziedziczący od T_1 i spełniający te same warunki, to wystąpienia te muszą być na poziomie T_1 agregowane. Formalnie można zapisać, że dla struktury regularnej:

$$\begin{aligned} \forall A_1, A_2 \in \mathbf{A}, A_1 \neq A_2 ((\exists T_1 \in \mathbf{T} (f_{inst}(A_1) \preceq T_1 \wedge f_{inst}(A_2) \preceq T_1 \wedge \\ \forall p \in f_P(T_1) f_V(A_1, p) = f_V(A_2, p)) \wedge \\ \neg \exists T_2 \in \mathbf{T} (T_2 \preceq T_1 \wedge f_{inst}(A_1) \preceq T_2 \wedge f_{inst}(A_2) \preceq T_2 \wedge \\ \forall p \in f_P(T_2) f_V(A_1, p) = f_V(A_2, p)) \rightarrow \\ (\exists A_3 (f_{inst}(A_3) = T_1 \wedge A_1 \ll A_3 \wedge A_2 \ll A_3))). \end{aligned} \quad (6.13)$$

■

Dowód:

Ponownie przeprowadzimy dowód nie wprost. Zakładamy, że wybieramy dowolne, ale ustalone wystąpienia kontekstów $A_1, A_2 \in \mathbf{A}$ i typ kontekstu T_1 , spełniających następujące warunki z poprzednika formuły (6.13):

$$f_{inst}(A_1) \preceq T_1 \wedge f_{inst}(A_2) \preceq T_1. \quad (6.14)$$

Zaprzeczeniem następnika będzie zdanie:

$$\neg \exists A_3 (f_{inst}(A_3) = T_1 \wedge A_1 \ll A_3 \wedge A_2 \ll A_3),$$

czyli A_1 i A_2 nie mają T_1 agregatu. Ponieważ również nie istnieje taki typ T_2 , że: $T_2 \trianglelefteq T_1$, $f_{inst}(A_1) \trianglelefteq T_2$, $f_{inst}(A_2) \trianglelefteq T_2$ i $\forall p \in f_P(T_2) f_V(A_1, p) = f_V(A_2, p)$, to A_1 i A_2 nie będą też posiadały agregatu na poziomie niższym niż T_1 . Z definicji 6.1 wynika, że wystąpienia te będą posiadały w T_1 różne rzuty: $A_1[T_1] \neq A_2[T_1]$, a to łącznie z (6.14) spełnia poprzednik zasady jednoznaczności regularnej wyrażonej formułą (6.11). Dla struktury regularnej musi więc być spełniony następnik (6.11), który mówi, że: $\exists p \in f_P(T_1) (f_V(A_1[T_1], p) \neq f_V(A_2[T_1], p))$. Na podstawie definicji 6.1 można to zdanie przekształcić na $\exists p \in f_P(T_1) (f_V(A_1, p) \neq f_V(A_2, p))$, a to zaprzecza członowi poprzednika (6.13) mówiącemu, że: $\forall p \in f_P(T_1) f_V(A_1, p) = f_V(A_2, p)$. QED

Struktura regularna zostanie ponownie wykorzystana w punkcie 6.1.4.

6.1.2 Schemat i instancje bazy wiedzy

Przyjęcie podziału aktywności na projektowe i eksploatacyjne zmusza do określenia, która część kontekstowej bazy wiedzy jest efektem pracy twórców systemu, a która powstaje w wyniku działań użytkowników. Należy, podobnie jak w metodzie relacyjnej, dokonać podziału bazy na schemat i instancję. Schemat jest wynikiem konceptualizacji i powinien ją w maksymalnym stopniu odzwierciedlać. W kontekstowej bazie wiedzy stworzonej zgodnie z metodą SIM wynikiem konceptualizacji jest kontekstowa terminologia wraz z porządkującą ją relacją dziedziczenia oraz przypisane jej aksjomaty. Z rozważań w poprzednim punkcie wynika, że istotny wpływ na konceptualizację mają miejsca agregacji wystąpień kontekstów. Dlatego schemat musi określić dopuszczalne miejsca agregacji dla każdego typu kontekstu. Można przyjąć uzgodnienie, że *dopuszczalnymi miejscami agregacji* dla typu kontekstu T są typy kontekstów, których wystąpienia mogą agregować wystąpienia typu T . Oczywiście, zgodnie z definicją 5.3, typy te muszą być typami dziedziczonymi przez T . Po przyjęciu takiego uzgodnienia można już zdefiniować, czym jest schemat, wydzielając z definicji kontekstowej bazy wiedzy odpowiednie fragmenty i dodając funkcję przyporządkowującą typom kontekstów dopuszczalne miejsca agregacji.

Definicja 6.2 (schemat kontekstowej bazy wiedzy wg metody SIM)

Schematem kontekstowej bazy wiedzy $\mathcal{K} = (S, f_{cont})$, gdzie $S = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \trianglelefteq, \ll, f_{inst})$, a $f_{cont}: \mathbf{T} \uplus \mathbf{A} \rightarrow \mathcal{P}(\mathcal{L})$, nazywamy trójkę $(S_T, f_{Tcont}, f_{agr})$, gdzie:

- 1) $S_T = (\mathbf{T}, T_{max}, \trianglelefteq)$ jest strukturą schematu,
- 2) $f_{Tcont}: \mathbf{T} \rightarrow \mathcal{P}(\mathcal{L}_T)$ jest *funkcją zawartości*, która przyporządkowuje typowi kontekstu fragment terminologii ontologii,
- 3) $f_{agr}: \mathbf{T} \rightarrow \mathcal{P}(\mathbf{T})$ jest funkcją, która przyporządkowuje każdemu typowi kontekstu dopuszczalne miejsca agregacji, taką że $\forall T \in \mathbf{T} (f_{agr}(T) \subseteq \{T' \mid T \trianglelefteq T'\})$.

■

Pozostałe składniki bazy wiedzy tworzą jej instancję. Instancja zawiera więc kontekstowy opis świata wraz z relacją agregacji, przypisane mu asercje oraz funkcję konkretyzacji. Ze względu na tę funkcję instancja musi znać swój schemat.

Definicja 6.3 (instancja kontekstowej bazy wiedzy wg metody SIM)

Instancją kontekstowej bazy wiedzy $\mathcal{K} = (S, f_{cont})$, gdzie $S = (\mathbf{T} \uplus \mathbf{A}, T_{max}, A_{max}, \trianglelefteq, \ll, f_{inst})$, a $f_{cont}: \mathbf{T} \uplus \mathbf{A} \rightarrow \mathcal{P}(\mathcal{L})$, nazywamy trójkę $(S_A, f_{inst}, f_{Acont})$, gdzie:

- 1) $S_A = (\mathbf{A}, A_{max}, \ll)$ jest strukturą instancji,
- 2) $f_{inst}: \mathbf{A} \rightarrow \mathbf{T}$ jest funkcją konkretyzacji,

- 3) $f_{Acont}: \mathbf{A} \rightarrow \mathcal{P}(\mathcal{L}_A)$ jest funkcją zawartości, przyporządkowującą wystąpieniu kontekstu fragment opisowi świata ontologii. ■

Rozdzielenie bazy wiedzy na schemat i instancję jest z matematycznego punktu widzenia odwracalne. Z elementów struktury schematu i instancji oraz z funkcji f_{inst} wziętej z krotki instancji można utworzyć strukturę bazy wiedzy, a funkcja f_{cont} jest po prostu sumą f_{Tcont} i f_{Acont} . Można jednak przyjąć inaczej, to znaczy założyć, że istnieje jakiś schemat H i jakaś instancja E , i zdefiniować kontekstową bazę wiedzy jako parę $\mathcal{K} = (H, E)$. Czyli zbiór wszystkich możliwych do utworzenia kontekstowych baz wiedzy $R_{\mathcal{K}}$ jest relacją $R_{\mathcal{K}} \subset \mathbf{H} \times \mathbf{E}$, gdzie \mathbf{H} i \mathbf{E} jest odpowiednio zbiorem wszystkich możliwych schematów i wszystkich możliwych instancji. Zawieranie się relacji $R_{\mathcal{K}}$ w iloczynie kartezjańskim jest właściwe, gdyż nie jest możliwe dowolne łączenie schematów z instancjami. Oto właściwości tej relacji, czyli lista warunków, które para (H, E) musi spełniać, aby być uznana za kontekstową bazę wiedzy. Niech $H = (S_T, f_{Tcont}, f_{agr})$, $E = (S_A, f_{inst}, f_{Acont})$, $S_T = (\mathbf{T}, T_{max}, \trianglelefteq)$ i $S_A = (\mathbf{A}, A_{max}, \ll)$. Para (H, E) może być elementem relacji $R_{\mathcal{K}}$ wtwg:

- 1) $\forall A \in \mathbf{A} (f_{inst}(A) \in \mathbf{T}) \wedge f_{inst}(A_{max}) = T_{max}$,
- 2) $\forall A, A' (A \ll A' \rightarrow f_{inst}(A') \in f_{agr}(f_{inst}(A)))$,
- 3) $\forall A, T (f_{inst}(A) = T \rightarrow (Sig_C(f_{Acont}(A)) \cup Sig_R(f_{Acont}(A)) \subseteq S(T))$.

Właściwość pierwsza nazywa się *odpowiedniością ze względu na konkretyzację* i mówi, że w kontekstowym opisie świata \mathbf{A} mogą znajdować się wystąpienia tylko tych typów, które należą do kontekstowej terminologii \mathbf{T} . Właściwość druga, *odpowiedniość ze względu na agregację*, mówi, że każde wystąpienie danego typu może być bezpośrednio agregowane tylko w dopuszczalnych miejscach agregacji dla tego typu. Odpowiedniość ze względu na konkretyzację i odpowiedzialność ze względu na agregację tworzą łącznie *odpowiedniość strukturalną*. Ostatnia właściwość nazywa się *odpowiedniością lingwistyczną* i oznacza, że wystąpienia danego typu nie mogą mieć przyporządkowanych asercji, które używają nazw konceptów i ról, które nie są obecne w sygnaturze tego typu.

Aby więc instancja E mogła utworzyć razem z danym schematem H kontekstową bazę wiedzy, musi odpowiadać mu strukturalnie i lingwistycznie. Instancję E nazywamy wtedy *instancją schematu H* . Instancje różnych schematów nie muszą być rozłącznymi podzbiorami zbioru \mathbf{E} . Wymienione warunki nie wykluczają, że dana instancja mogłaby być instancją więcej niż jednego schematu. Podzbiór \mathbf{E} zawierający wszystkie instancje danego schematu H , to znaczy wszystkie instancje odpowiadające mu strukturalnie i lingwistycznie, oznaczany będzie literą \mathbf{E}_H . Zdefiniowanie bazy wiedzy jako pary (H, E) utrudnia dostęp do poszczególnych ich składowych oraz do elementów ich struktur. Wygodne będzie przyjęcie umowy upraszczającej oznaczenia:

- $H[\mathbf{T}]$ — terminologia kontekstowa,
- $H[\trianglelefteq]$ — relacja dziedziczenia,
- $H[T_{max}]$ — element maksymalny relacji dziedziczenia,
- $H[f_{Tcont}]$ — funkcja zawartości dla terminologii,
- $H[f_{agr}]$ — funkcja przyporządkowująca dopuszczalne miejsca agregacji.

Analogicznie dla instancji: $E[\mathbf{A}]$, $E[\ll]$, $E[A_{max}]$, $E[f_{inst}]$, $E[f_{Acont}]$, to odpowiednio: kontekstowy opis świata, relacja agregacji, maksymalny element agregacji, funkcja konkretyzująca i funkcja zawartości dla \mathbf{A} .

Dla porządku warto jeszcze w tym miejscu powiedzieć, że wprowadzenie ograniczenia na dopuszczalne miejsca agregacji dla danego typu zmienia nieco pewne ustalenia dotyczące struktur regularnych: typy T_1 i T_2 z definicji 6.1 (rzutu regularnego) oraz z twierdzenia 6.1 (o konieczności agregatu) muszą podlegać temu ograniczeniu, czyli należeć do dopuszczalnych miejsc agregacji dla wystąpień, o których tam mowa. Nie zmienia to ich sensu, gdyż chodzi w nich o agregowanie danych wystąpień w najniższym *możliwym* miejscu, a ograniczenie, o którym mowa, zmniejsza jedynie liczbę możliwości.

W kolejnym punkcie zostanie podjęta próba naszkicowania sposobu eksploatacji systemu zgodnego z metodą SIM.

6.1.3 Opis działania systemu zgodnego z SIM

We wstępie do niniejszej pracy autor dał wyraz swojemu przekonaniu, że poprawnie sformułowana metoda tworzenia ontologii kontekstowych mogłaby otworzyć drogę dla nowej klasy systemów informatycznych. Przypominałyby one w dużo większym stopniu obecnie projektowane systemy powszechnego użytku niż typowe systemy wnioskujące. W tym punkcie omówionych jest kilka aspektów dotyczących ewentualnego sposobu funkcjonowania takich systemów.

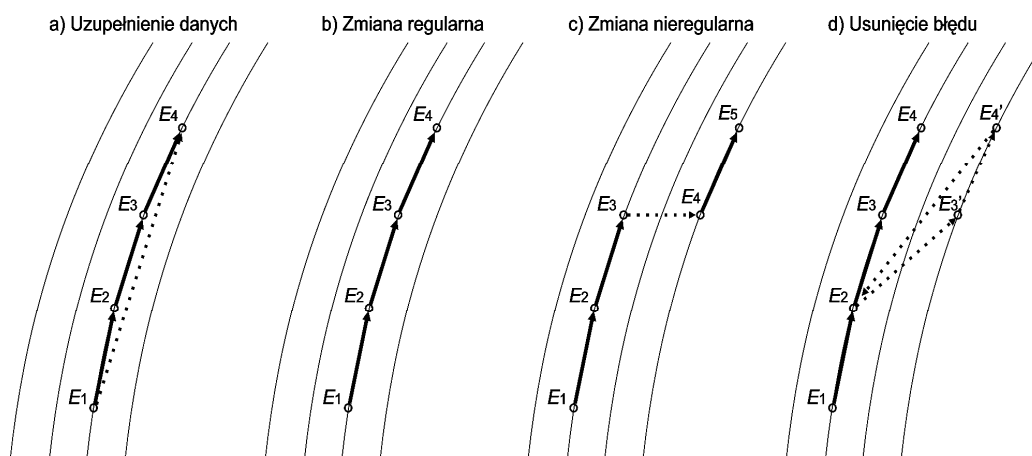
Zgodnie z tym, co zostało powiedziane w poprzednim punkcie, system gotowy do rozpoczęcia eksploatacji zawierałby jedynie schemat. Działania użytkownika byłyby podzielone na dwa tryby: wprowadzanie danych oraz zadawanie zapytań. Wprowadzanie danych polegałoby na zmianie instancji bazy wiedzy: następowałoby przejście od instancji E_i do instancji E_{i+1} . Wiązałoby się to z dodawaniem nowych wystąpień kontekstów oraz na modyfikacji lub usuwaniu istniejących. Modyfikacja lub usunięcie powinno być oczywiście poprzedzone wyborem wystąpienia kontekstu, natomiast dodanie nowego wystąpienia powinno być poprzedzone wyborem typu kontekstu oraz miejsca jego agregacji spośród dopuszczalnych miejsc zdefiniowanych dla wybranego typu. Dokonywanie tego wyboru może być nieświadome (tak jak w systemach relacyjnych, użytkownik nie musi koniecznie wiedzieć, która z relacji jest modyfikowana na skutek jego działań). Dużo łatwiej zapewne, niż wybrać kontekst, będzie użytkownikowi określić pewne wartości parametrów kontekstowych, co pozwoli na automatyczne wybranie zarówno typu, jak i miejsca agregacji kontekstu.

Kolejne zmiany wprowadzane do bazy budują ciąg instancji $c_E = (E_1, \dots, E_n)$. W metodzie relacyjnej przyjmuje się, że dana instancja bazy danych tworzy jej chwilowy *stan*. Po zmianie stanu jest to wciąż ta sama baza. W przypadku baz wiedzy przyjęcie podobnej umowy nazewnicznej wymaga dodatkowego omówienia. Wynika to z faktu, że zmieniając instancję użytkownik kieruje się różnymi intencjami. Można wyodrębnić kilka typów zmian, które różnią się znaczeniem zmiany:

- *Uzupełnienie danych* — z semantycznego punktu widzenia nie jest to zmiana. Wprowadzenie takiego zapisu jest spowodowane tym, że z jakichś przyczyn (najczęściej z powodu objętości danych) użytkownik nie był w stanie wpisać wszystkich danych w jednym kroku. Z punktu widzenia bieżących rozważań najlepiej byłoby uznać, że taka zmiana nie tworzy nowej instancji. Monotoniczność zostaje zachowana.
- *Zmiana regularna* — jest to zmiana, która podlega regułom narzuconym przez projektanta i dlatego da się opisać parametrami kontekstowymi. Również w tym przypadku monotoniczność bazy jest zachowana.

- *Zmiana nieregularna* — jest to zmiana, która nie podlega regułom narzuconym przez projektanta, więc nie da się opisać za pomocą parametrów kontekstowych. To, że system taką zmianę umożliwia, jest wynikiem scedowania na użytkownika niektórych decyzji konceptualizacyjnych. Użytkownik odstąpił od ścisłego trzymania się reguł („instrukcji obsługi”), wyobraził sobie jakiś nowy parametr kontekstowy i właśnie zmiana jego wartości spowodowała konieczność zweryfikowania bazy wiedzy. Nie powstaje nowy kontekst. Zmiana narusza zasadę monotoniczności.
- Usunięcie błędu — jest właściwie pewnym rodzajem zmiany nieregularnej, gdyż z pewnością pociąga za sobą utratę monotoniczności. Jednak z punktu widzenia rozważań teoretycznych wygodnie jest przyjąć, że nie jest to zmiana. Można po prostu uznać, że jest to „cofnięcie się w czasie”, czyli usunięcie jednego lub kilku ostatnich elementów ciągu c_E i ponowne ich dodanie, tym razem bezbłędne.

Wymieniony typy zmian zostały przedstawione w postaci ciągów c_E na rysunku 6.1. Zmiany regularne, opisane odpowiednimi parametrami i zachowujące monotoniczność bazy przedstawione są za pomocą odcinków narysowanych linią ciągłą. Zmiany, które nie mieszczą się w parametrycznym opisie kontekstowym, a które powinny się mieścić, gdyby odbywały się zgodnie z regułami, oznaczone są odcinkami narysowanymi linią przerywaną. Cienkimi, równoległymi łukami zaznaczone są „ścieżki”, którymi powinien postępować rozwój bazy wiedzy, gdyby miała ona zachować monotoniczność. Trzymając się tej ścieżki użytkownik opisuje pewien zbiór światów możliwych, który został przez projektantów przewidziany w procesie konceptualizacji. Dzięki temu system posiada odpowiednie środki do opisu tego zbioru. Przeskok z łuku na łuk symbolizuje przeskok z jednego do drugiego świata możliwego. System nie posiada środków do opisu takiego przejścia.



Rys. 6.1 Sposób przechodzenia od instancji do instancji dla różnych typów zmian.

Dwa pierwsze rodzaje zmian można uznać za zmiany stanu danej bazy wiedzy. Różnica między nimi jest trudna do zdefiniowania, ponieważ trudno jest dostrzec różnicę między sytuacją, w której użytkownik wpisuje coś do bazy, gdyż wczoraj nie zdążył tego zrobić, a sytuacją, w której użytkownik robi to, gdyż przed chwilą dowiedział się o nowych faktach.

Dwa ostatnie rodzaje zmian nie są zmianą stanu — są to sytuacje, w których jest inicjowana nowa baza wiedzy. Usunięcie błędu jest, jak to zostało nazwane, „cofnięciem się w czasie”. Należy uściślić, o jakie błędy chodzi. Nie chodzi tu o błędy kontrolowane przez system, na przykład błędy polegające na niewłaściwym wystawieniu faktury przez pracownika zajmującego się sprzedażą. Najczęściej w systemach księgowych błędnie wystawionego dokumentu nie da się już wycofać i konieczne jest wyemitowanie korekty. Nie

są to w takim razie błędy w rozumieniu logiki biznesowej. Błędy, o jakich tu mowa, nie są przewidziane przez twórców systemu. Rzadko dają się naprawić z wnętrza systemu, częściej należy użyć narzędzi pozasystemowych lub specjalnych środków dostarczonych przez producenta systemu (na przykład narzędzia podnoszące bazę po awarii). Usunięcie takiego błędu polega więc na „oszukaniu” systemu poprzez zlikwidowanie śladu po błędnie wprowadzonych lub uszkodzonych danych i wprowadzeniu danych bezbłędnych. Nawet jeżeli system pozwala na tego rodzaju działania, nie należy tego rodzaju zmiany rozpatrywać jako zmiany stanu bazy, lecz raczej jako korektę większego fragmentu ciągu c_E .

Zmiana nieregularna, mimo utraty monotoniczności przez bazę wiedzy, nie jest zjawiskiem niekorzystnym. Możliwość wprowadzania takich zmian w systemie świadczy o tym, że twórcy, jak to już zostało powiedziane, zdecydowali się scedować część swoich kompetencji na użytkownika. Jeśli, na przykład, system pozwala na wstawienie asercji `Student(jan_kowalski)`, a następnie na jej usunięcie (gdy ktoś był przez jakiś czas studentem, a potem przestał nim być), jeśli da się zmienić czyjś adres zamieszkania bez zachowania adresu poprzedniego (gdy ktoś się przeprowadził), oznacza to, że projektant ani nie dołączył parametru kontekstowego mówiącego, kiedy dana informacja jest ważna, ani nie wprowadził odpowiednich środków ontologicznych (konceptów, ról), by to wyrazić. Jest to więc luka w konceptualizacji, którą musi wypełnić użytkownik, by sprawić, aby system był dla niego użyteczny. W praktyce bardzo wiele systemów użytkowych w mniejszym lub większym stopniu pozwala naruszyć zasadę monotoniczności. Niesie to za sobą pewne niedogodności (na przykład stare wydruki zestawień stają się nieaktualne), lecz jeśli nie jest to system wnioskujący, od którego wymaga się aby był pełny i poprawny, nie stanowi to większego problemu. Inaczej jest w przypadku systemów, w których wnioskowanie ma miejsce. Utrata monotoniczności unieważnia wszelkie wnioski, które zostały do tej pory wyciągnięte. Choć niektóre systemy ontologiczne zezwalają na usuwanie asercji, na przykład język DIG 2.0 (<http://dig.cs.manchester.ac.uk/overview.html>) obsługiwany przez takie systemy wnioskujące, jak Racer lub FaCT, to jednak jest to interpretowane raczej jako usunięcie błędu (wycofanie, odwołanie — ang. *retract*), niż zezwolenie na wprowadzenie niemonotonicznej zmiany do ontologii. Z punktu widzenia obecnych rozważań taka zmiana jest także znaczącym utrudnieniem. Zaniechanie przez projektantów dokonania precyzyjnej konceptualizacji wywołuje zjawisko, które można nazwać *nieoznaczonością konceptualizacyjną*. Trudno jest odnieść się do tego zjawiska w rozważaniach, które dotyczą właśnie konceptualizacji. Dlatego należy przyjąć, że wprowadzenie zmiany nieregularnej powoduje przerwanie ciągu instancji (stanów) bazy wiedzy i rozpoczęcie nowego, czyli następuje zainicjowanie nowej bazy wiedzy.

Nieoznaczoność konceptualizacyjna jest, w przypadku zastosowania metody SIM, znacznie łatwiejsza do uniknięcia. W ramach tej metody projektant dysponuje środkami, które pozwalają aktualizować dane bez usuwania starych — zaktualizowane dane mogą być po prostu zapamiętane obok starych w nowym wystąpieniu kontekstu z nową wartością parametru kontekstowego odzwierciedlającego upływ czasu.

Drugi tryb działania użytkownika — zadawanie zapytań — nie musi się znacznie różnić od zadawania zapytań do znanych obecnie systemów wnioskujących. Dotyczyłyby one sprawdzenia spełnialności konceptów (ról), określenia, czy dany osobnik (para osobników) jest wystąpieniem danego konceptu (roli), wymienienia wszystkich znanych wystąpień danego konceptu (roli). Należy jednak zwrócić uwagę na dwa bardzo ważne problemy. Pierwszy, to kontekstowość zapytań. Zgodnie z regułą lokalności z rozdziału 4.1.1 wyprowadzanie formuł (lub ich wartościowanie) musi zachodzić w jakimś kontekście. To wcale nie oznacza, że kontekstem zapytania musi być kontekst istniejący w ramach bazy wiedzy. Można sobie wyobrazić, że kontekst zapytania jest tworzony *ad hoc* wyłącznie dla

potrzeb danego zapytania poprzez jakieś przetworzenie kontekstów istniejących. To jednak doprowadza do dyskusji na temat algebry kontekstów i odpowiedniego języka zapytań.

Drugim problemem, który w przyszłości powinien zostać rozwiązany, jest mała kompleksowość zapytań możliwych do sformułowania w obecnie proponowanych językach ontologicznych. W tradycyjnych systemach informatycznych odpowiedzi na zadawane zapytania przybierają formę raportów przedstawiających użytkownikowi pewien kompleks zinterpretowanych już danych. Aby takie raporty uzyskać w systemie współpracującym z bazą wiedzy, należałoby albo wykonać bardzo wiele zapytań i odpowiednio połączyć ich wynik w jeden sensowny raport, albo zastanowić się nad zdefiniowaniem nowego języka pozwalającego na zadawanie zapytań dających gotowy, kompleksowy wynik.

Jak widać rozwiązaniem obu problemów jest nowy język zapytań. Temat ten zostanie przedyskutowany w rozdziale 8. W tym momencie należy założyć, że kontekstem zapytania będzie jeden z kontekstów już istniejących i że zapytanie nie będzie go zmieniało. Na przykład nie będzie mogła być zmieniona sygnatura kontekstu. Oznacza to, że zapytanie zawierające symbole pozalogiczne nie należące do sygnatury kontekstu (nazwy osobników, konceptów lub ról nie wprowadzone wcześniej) nie będzie mogło być wykonane.

Wymienione dwa tryby działań użytkownika są jedynie pewnym modelem formalnym. W praktyce łatwo jest wyobrazić sobie, że czynności eksploatacyjne mogą także obejmować zmiany terminologiczne w bazie wiedzy. Możliwe jest nawet stworzenie systemu „samouczącego się”, czyli takiego, w którym dołączanie nowych pojęć, nowych typów kontekstów i miejsc agregacji wystąpień będzie odbywało się automatycznie. Jednak na potrzeby rozważań teoretycznych, także rozważań prowadzonych w niniejszej rozprawie, należy przyjąć zaproponowane powyżej uproszczenie, traktując inne modele jako przypadki szczególne.

6.1.4 Metoda OntoClean a hierarchiczna kontekstualizacja bazy wiedzy

Po określeniu roli parametrów kontekstowych w procesie projektowania kontekstowej bazy wiedzy oraz w trakcie jej eksploatacji, możliwe staje przeanalizowanie wpływu cech metody SIM na proces konceptualizacji. Jak stwierdzono w podrozdziale 3.3, bardzo wartościowym narzędziem wspomagającym pracę projektanta jest metoda walidacji ontologii OntoClean (opisana w podrozdziale 3.1). Metoda ta, w zastosowaniu do ontologii niezmodularyzowanych, ma za zadanie uchronić je przed zbyt głębokimi zmianami relacji subsumcji w sytuacji, gdy zmiana opisywanego stanu rzeczy wymusi wprowadzenie zmian niemonotonicznych. Skonfrontowanie metody OntoClean z metodą hierarchicznej kontekstualizacji SIM daje zupełnie nowe spojrzenie na problem metasystematyzacji konceptualnej. Metoda SIM, dzięki swoim właściwościom, pozwala uniknąć niemonotonicznych zmian w sytuacji zmiany punktu widzenia. Gdy zatem jakieś zdanie straci swoją ważność (na przykład ktoś przestanie studiować), nie ma potrzeby usuwania nieaktualnej asercji (na przykład $Student(jan_kowalski)$) i wprowadzania na jej miejsce nowej, mówiącej coś przeciwnego (na przykład asercji $\neg Student(jan_kowalski)$). Można po prostu dodać nowy kontekst. Jednakże nie oznacza to, że reguły metody OntoClean utraciły swoje znaczenie. Przeciwnie, mogą one stać się narzędziem wspomagającym projektowanie poprawnej struktury kontekstowej bazy wiedzy. Dodatkowo, dzięki uzależnieniu metasystematyzacji konceptualnej od kontekstu, zasady metody OntoClean nie niosą ze sobą niebezpieczeństwa usztywnienia punktu widzenia, gdyż będą dotyczyć tylko wydzielonej grupy kontekstów.

Wpływ, jaki reguły metody OntoClean powinny wywierać na decyzje związane z projektowaniem struktury kontekstowej bazy wiedzy, zostanie przedstawiony poprzez omówienie jednej metawłaściwości — immanentności.

Immanentność jest zdefiniowana w punkcie 3.1.1.1. Przydatne będzie przypomnienie formuły, która ją opisuje: właściwość φ jest *immanentna* wtwg $\forall x (\varphi(x) \rightarrow \Box \varphi(x))$. Autorzy metody OntoClean posłużyli się *kwantyfikowaną logiką modalną*, aby wyrazić, że dla każdej encji x , jeśli φ jest właściwością tej encji, to jest jej właściwością *koniecznie*, czyli w każdym świecie możliwym. Takie ujęcie jest możliwe, gdyż w kwantyfikowanej logice modalnej [G2006] przyjmuje się tzw. *sztynny desygnator*. W tym przypadku oznacza to, że nazwa x jest tak samo interpretowana (wskazuje ten sam obiekt) we wszystkich światach możliwych. Obiekt ten, gdy raz zostanie mu przypisana dana właściwość, na przykład *bycie człowiekiem*, nigdy, czyli w żadnym świecie możliwym, tej właściwości nie straci.

Jak zostało powiedziane w podrozdziale 3.3, osoby aktualizujące daną ontologię, ponownie ją wykorzystujące lub dokonujące jej integracji z innymi ontologiami wprowadzają do niej zmiany, które mogą być związane ze zmianą opisywanego stanu rzeczy (świata możliwego) lub zmianą punktu widzenia na pewne zagadnienia. Nowy stan rzeczy lub nowy punkt widzenia tworzą nowy kontekst opisu, który, zgodnie z przyjętą w tej pracy za podstawę metaforą pudełka, da się scharakteryzować za pomocą zbioru parametrów kontekstowych. W przypadku ontologii niezmodularyzowanych, nie istnieje zwyczaj wyszczególniania takich parametrów, gdyż ontologia taka nie jest traktowana jako kontekstowa. Jednakże, patrząc na dłuższy odcinek życia ontologii, nie można zaprzeczyć, że kolejne jej wersje są związane ze zmianami w obrębie wymiarów, które przyjęto jako charakterystyczne dla opisu kontekstu (zakresu, poziomu szczegółowości, perspektywy). Istnieje więc jakiś zbiór parametrów kontekstowych $P = \{p_1, \dots, p_n\}$, który, gdy parametrom tym zostaną nadane wartości, opisuje poszczególne wersje ontologii. Dla ogólności rozważań można przyjąć, że zbiór ten jest nieograniczony, czyli że $n \rightarrow \infty$. Dopuszczalne wartości tych parametrów tworzą relację $\bar{V}_P \subset \prod_{p \in P} V_p$. Z pewnością nie cała przestrzeń $\prod_{p \in P} V_p$ jest dostępna, gdyż niektóre parametry są od siebie nawzajem zależne. Każdy element tej relacji, $\bar{v} = \{(p_1, v_1), \dots, (p_n, v_n)\}$ odpowiada jednemu światu możliwemu. Opisanie kontekstowego tła danej wersji ontologii O polega na przypisaniu jej jednego elementu relacji \bar{V} poprzez nadanie wartości parametrom kontekstowym¹⁸. Wyjaśnienia wymaga jeszcze termin „wersje ontologii”. Na ogół nie ma zgody co do tego, czy wprowadzenie zmiany do ontologii (szczególnie zmiany niemonotonicznej) jest zmianą wersji tej ontologii, czy raczej utworzeniem nowej ontologii. W tych rozważaniach przyjęto, że zmiany nie powodują powstania nowej ontologii. Każda zmiana w ontologii O (chodzi przede wszystkim o zmiany nazwane w punkcie 6.1.3 nieregularnymi, gdyż zmiany regularne można uznać za uzupełnianie danych) jest związana ze zmianą kontekstu, czyli z przemieszczeniem się opisu kontekstowego tła z punktu opisanego krotką \bar{v} do punktu opisanego krotką \bar{v}' (gdzie $\bar{v}, \bar{v}' \in \bar{V}$) oraz z przejściem ontologii O z wersji $O(\bar{v})$ do wersji $O(\bar{v}')$.

Przejście od koncepcji światów możliwych do koncepcji parametrów opisujących tło kontekstowe kolejnych wersji ontologii pozwala inaczej zdefiniować pojęcie immanentności.

¹⁸ Dla uściślenia: jedna krotka relacji \bar{V}_P o nieskończonej liczbie elementów (reprezentująca jeden świat możliwy) opisuje daną ontologię, ale nie istnieje odwrotna zależność. Ontologia jest nieprecyzyjnym opisem świata (patrz punkt 2.1.4) i można jej przypisać cały zbiór światów możliwych (cały zbiór krotek relacji \bar{V}_P) mogących być jej interpretacją.

Definicja 6.4 (immanentność konceptu)

Dany koncept C jest *immanentny* w ontologii O wtwg

$$\forall a (\exists \bar{v} \in \bar{V}_P O(\bar{v}) \models C(a) \rightarrow \forall \bar{v}' \in \bar{V}_P (a \in \text{Sig}(O(\bar{v}')) \rightarrow O(\bar{v}') \models C(a))). \quad \blacksquare$$

W powyższej definicji wprowadzono już pojęcia znane z ontologii wyrażonych w logice opisowej. Tak więc pojęcie właściwości zostało zastąpione pojęciem konceptu, immanentność konceptu dotyczy konkretnej ontologii O , a a jest nazwą osobnika zawartą w sygnaturze tej ontologii. Jednak sens definicji jest taki sam. Mówiąc językiem naturalnym, koncept C jest immanentny w ontologii O , jeśli istnienie jakiejś wersji tej ontologii, w której dany osobnik jest wystąpieniem konceptu C implikuje, że jest on wystąpieniem tego konceptu we wszystkich tych wersjach tej ontologii, których sygnatura zawiera nazwę tego osobnika.

Powyższa forma definicji 6.4 ukazuje analogię do definicji immanentności z punktu 3.1.1.1., ale nie jest wygodna do stosowania w dalszych rozważaniach. Dlatego należy przekształcić ją do postaci:

$$\forall a (\forall \bar{v}, \bar{v}' \in \bar{V}_P (a \in \text{Sig}(O(\bar{v})) \cap \text{Sig}(O(\bar{v}')) \rightarrow (O(\bar{v}) \models C(a) \leftrightarrow O(\bar{v}') \models C(a)))) \quad (6.15)$$

Powyższa formuła jest równoważna formule zaprezentowanej w definicji.

Dowód:

Z definicji po przeniesieniu kwantyfikatora $\forall \bar{v}' \in \bar{V}$ na początek formuły:

$$\forall a (\forall \bar{v} \in \bar{V}_P (\forall \bar{v}' \in \bar{V}_P (O(\bar{v}) \models C(a) \rightarrow (a \in \text{Sig}(O(\bar{v}')) \rightarrow O(\bar{v}') \models C(a))))))$$

Po dodaniu $a \in \text{Sig}(O(\bar{v}))$ do poprzednika pierwszej implikacji (zdanie $C(a)$ nie może wynikać z ontologii, jeśli a nie znajduje się w sygnaturze, czyli w słowniku):

$$\forall a (\forall \bar{v}, \bar{v}' \in \bar{V}_P (a \in \text{Sig}(O(\bar{v})) \wedge O(\bar{v}) \models C(a) \rightarrow (a \in \text{Sig}(O(\bar{v}')) \rightarrow O(\bar{v}') \models C(a))))$$

Po zastosowaniu równoważności $(a \wedge b) \rightarrow (c \rightarrow d) \equiv (c \wedge b) \rightarrow (a \rightarrow d)$ i zgodnym z nią przekształceniu powyższej formuły otrzymujemy:

$$\forall a (\forall \bar{v}, \bar{v}' \in \bar{V}_P (a \in \text{Sig}(O(\bar{v})) \wedge a \in \text{Sig}(O(\bar{v}')) \rightarrow (O(\bar{v}) \models C(a) \rightarrow O(\bar{v}') \models C(a))))$$

W powyższej formule drugą implikację można zastąpić równoważnością, ponieważ \bar{v} i \bar{v}' są nierozróżnialne (można zamienić je miejscami). Dodatkowo można wyrażenie $a \in \text{Sig}(O(\bar{v})) \wedge a \in \text{Sig}(O(\bar{v}'))$ wymienić na równoważne $a \in \text{Sig}(O(\bar{v})) \cap \text{Sig}(O(\bar{v}'))$. Stąd:

$$\forall a (\forall \bar{v}, \bar{v}' \in \bar{V}_P (a \in \text{Sig}(O(\bar{v})) \cap \text{Sig}(O(\bar{v}')) \rightarrow (O(\bar{v}) \models C(a) \leftrightarrow O(\bar{v}') \models C(a))))),$$

czyli zdanie (6.15), której równoważność z formułą z definicji 6.4 należało udowodnić.

QED

Wprowadzenie ontologii kontekstowych pozwala na objęcie konceptualizacją różnych punktów widzenia, a więc różnych światów możliwych. Można powiedzieć, że jedna ontologia kontekstowa obejmuje swoim opisem całą „wiązkę” światów możliwych. Rozważając więc możliwe struktury ontologii kontekstowych, warto zadać pytanie, jak różne parametry (lub grupy parametrów) wpływają na metawłaściwości konceptów. Gdyby istniał jeden tylko wybór między globalną immanentnością a globalną (anty-) nieimmanentnością, to nadal obowiązywałaby definitywność ustalonego punktu widzenia. Dla ontologii konteksto-

wych globalna immanentność jest mniej interesująca niż immanentność lokalna. Warto więc taki rodzaj immanentności zdefiniować.

Definicja 6.5 (immanentność lokalna względem podzbioru parametrów)

Dany koncept C jest *immanentny lokalnie w ontologii O względem podzbioru parametrów kontekstowych P' ($P' \subset P$)* wtwg

$$\forall a (\forall \bar{v}, \bar{v}' \in \bar{V}_P ((a \in \text{Sig}(O(\bar{v})) \cap \text{Sig}(O(\bar{v}')) \wedge \bar{v}[P'] = \bar{v}'[P']) \rightarrow (O(\bar{v}) \models C(a) \leftrightarrow O(\bar{v}') \models C(a))))).$$

gdzie $\bar{v}[P'] = \{(p, v) \mid (p, v) \in \bar{v} \wedge p \in P'\}$. ■

Taki rodzaj immanentności jest dużo bardziej przydatny. Immanentność lokalna to immanentność w podzbiorze światów możliwych. Podzbiór ten charakteryzuje się tym, że parametry kontekstowe opisujące światy w nim zawarte mają w zakresie parametrów należących do P' ustaloną wartość. Zmiana wartości ustalonych może zmienić przypisanie osobnika do ekstensji konceptu C , ale nie sprawi, że koncept przestanie być immanentny lokalnie. Immanentność globalna jest w tym ujęciu specyficznym przypadkiem — jest to immanentność lokalna względem $P' = \emptyset$.

Operowanie pojęciem immanentności lokalnej pozwala w znacznym stopniu uniknąć wad normalizacji ontologii wymienionych w podrozdziale 3.3 i uelastyczyć ich projektowanie. Metoda SIM posiada środki, dzięki którym przy pewnych założeniach da się wyrazić immanentność lokalną i zagwarantować ją w trakcie eksploatacji. Założenia te określają sposób przejścia od opisywanej dotąd sytuacji ogólnej, abstrahującej od jakichś rozwiązań konkretnych, do sytuacji, która jest wyznaczona przez rozwiązania narzucone przez reguły metody SIM.

Pierwsze założenie dotyczy ujawnienia części parametrów kontekstowych. Dzięki temu ontologia kontekstowa obejmuje swoim opisem całą „wiązkę” światów możliwych. Niech od tego momentu symbol P_{max} oznacza skończony zbiór ujawnionych parametrów, a symbol $\bar{V}_{P_{max}}$ oznacza relację (skończonego stopnia) reprezentującą wszystkie dopuszczalne wartości tych parametrów. Przejście od nieskończonego zbioru parametrów kontekstowych do zbioru skończonego powoduje, że należy nieco zmienić wyobrażenie na temat związku między dopuszczalnymi wartościami tych parametrów a światami możliwymi. Jedną z konsekwencji jest to, że nie jest już możliwe tak samo precyzyjne opisywanie światów możliwych, przez co jeden zestaw wartości parametrów nie opisuje jednego świata możliwego, lecz pewien ich podzbiór. Pojawia się więc możliwość różnych interpretacji. Jest to jednak oczywiście akceptowalne, o ile wyznaczane podzbiory światów możliwych nie zawierają takich światów, które są sprzeczne z intencją twórców ontologii.

Choć rola parametrów kontekstowych nie jest przez formalizm metody ujęta, to jednak ich wpływ znajduje swoje odzwierciedlenie w strukturze utworzonej bazy wiedzy. W punkcie 6.1.1 został opisany sposób, w jaki należy stosować parametry kontekstowe podczas projektowania i jaki jest związek między strukturą bazy a przyjętym kontekstowym opisem parametrycznym. Z opisu tego wynika, że dana kontekstowa baza wiedzy nie ma przypisanego jednego zbioru parametrów, lecz każdy typ kontekstu posiada swój zbiór parametrów przyporządkowywany mu funkcją $f_P: \mathbf{T} \rightarrow \mathcal{P}(P_{max})$. Każdy ze zbiorów parametrów $P = f_P(T)$ wyznacza swój zbiór dopuszczalnych wartości, oznaczany $\bar{V}_P \subseteq \prod_{p \in P} V_p$, lub $\bar{V}_T \subseteq \prod_{p \in f_P(T)} V_p$ (tym razem zawieranie się może być niewłaściwe, gdyż w szczególnych przypadkach wszystkie ujawnione parametry mogą być od siebie niezależne).

Można przyjąć, że zawartość typu kontekstu T wyznacza pewną ontologię (lokalną) O_T , której wersje $O_T(\bar{v})$, gdzie $\bar{v} \in \bar{V}_T$, są związane z różnymi wartościami parametrów $p \in f_P(T)$. Wersje te są tworzone poprzez dołączanie kolejnych wystąpień typu T . Dołączenie

nowego wystąpienia, czyli utworzenie nowej wersji ontologii lokalnej, nie wiąże się z zerwaniem monotoniczności, gdyż, w przeciwieństwie do opisaney wyżej sytuacji dla płaskiej ontologii, każda tworzona wersja jest zapamiętywana. Jeśli użytkownik stosuje się do zaleceń projektanta i trzyma się jedynie tych światów możliwych (podzbiorów światów możliwych), które są opisywane przez narzucone w procesie projektowania parametry, będzie dokonywał w bazie wiedzy jedynie zmian regularnych. Wprowadzenie zmiany nieregularnej, czyli usunięcie jakiejś asercji lub jakiegoś wystąpienia, jest zerwaniem monotoniczności. Oznacza to, że użytkownik wyszedł poza narzucony zbiór parametrów. Drugim założeniem związanym z przejściem od sytuacji ogólnej do kontekstowej bazy wiedzy utworzonej w zgodzie z zasadami metody SIM jest więc przyjęcie, że wszelkie zmiany w bazie ograniczony są jedynie do przypadków, w których jest zachowana monotoniczność.

W ramach ostatniego założenia przyjmuje się, że zachowane zostaną zasady określone w punkcie 6.1.1 dla struktur regularnych.

Przyjęcie opisanych wyżej założeń pozwala zdefiniować immanentność lokalną dla metody SIM. Zanim jednak ta definicja zostanie sformułowana należy wprowadzić kilka dodatkowych oznaczeń. Niech:

- dla danego schematu H , zdefiniowanego w nim typu kontekstu T i dowolnej instancji $E \in \mathbf{E}_H$ symbol $E[\mathbf{A}_T]$ oznacza wszystkie wystąpienia kontekstu, takie że:
 $E[\mathbf{A}_T] = \{A \mid A \in E[\mathbf{A}] \wedge E[f_{inst}](A) = T\}$,
- \bar{f}_v oznacza funkcję, która każdemu wystąpieniu kontekstu $A \in \mathbf{A}$ przypisuje wartości wszystkich obowiązujących dla tego wystąpienia parametrów:
 $\text{Fnc}(\bar{f}_v) \wedge \bar{f}_v = \{(A, r) \mid r = \{(p, v) \mid p \in f_p(f_{inst}(A)) \wedge v \in V_p\}\}$,
- $\bar{f}_v(A)[P']$ oznacza przycięcie wartości funkcji $\bar{f}_v(A)$ do zbioru tylko tych elementów r , dla których $p \in P'$.

Immanentność lokalna jest wyżej definiowana dla danej ontologii O , więc tu będzie chodziło o $O_T = \text{def}(T)$, gdzie T jest danym typem kontekstu w danym schemacie H . Wszystkie możliwe wersje ontologii O_T oznaczają w nomenklaturze metody SIM wszystkie możliwe konteksty typu T , a te są wyznaczane przez wszystkie możliwe wystąpienia typu T we wszystkich możliwych instancjach schematu H . Tak więc zapis $\forall \bar{v} \in \bar{V}_T O_T(\bar{v})$ jest równoważny zapisowi $\forall E \in \mathbf{E}_H \forall A \in E[\mathbf{A}_T] \kappa = (T, A)$. Formułę z definicji 6.5 można więc przekształcić na symbole związane z metodą SIM w sposób następujący:

$$\forall a (\forall E, E' \in \mathbf{E}_H (\forall A \in E[\mathbf{A}_T] (\forall A' \in E'[\mathbf{A}_T] \\ ((a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \bar{f}_v(A)[P'] = \bar{f}_v(A')[P']) \rightarrow \\ (A \models C(a) \leftrightarrow A' \models C(a))))))$$

Jeśli, jak to zostało przyjęte wyżej, zmiany w obrębie bazy wiedzy ograniczą się jedynie do zmian regularnych, to możliwe jest uproszczenie kwantyfikatorów w ten sposób, aby dotyczyło ono wystąpień kontekstów z jednej tylko instancji. Zmiany regularne zakładają bowiem, że w ciągu instancji symbolizującym zmianę stanów jednej bazy wiedzy nie mogą się pojawić instancje łamiące zasadę monotoniczności. Definicja immanentności lokalnej dla metody SIM brzmi więc następująco:

Definicja 6.6 (immanentność lokalna w SIM względem podzbioru parametrów)

Koncept C jest *immanentny dla typu kontekstu T względem podzbioru parametrów P'* , gdzie $T \in H[\mathbf{T}]$ i $P' \subset f_p(T)$, wtwg

$$\forall a (\forall E \in \mathbf{E}_H (\forall A, A' \in E[\mathbf{A}_T] ((a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \bar{f}_v(A)[P'] = \bar{f}_v(A')[P']) \rightarrow \\ (A \models C(a) \leftrightarrow A' \models C(a))))). \quad \blacksquare$$

Mówiąc językiem naturalnym, definicja 6.6 określa koncept C immanentnym lokalnie dla typu kontekstu T w schemacie H względem danego zbioru parametrów kontekstowych P' , będących podzbiorem zbioru parametrów przypisanych typowi T , jeśli dla każdej pary wystąpień tego typu posiadających w swojej sygnaturze nazwę osobnika a oraz równe wartości przypisane parametrom z P' , w obu wystąpieniach pary jednocześnie a albo jest wystąpieniem konceptu C , albo nim nie jest.

W formule z definicji 6.6 warto zastąpić funkcję \bar{f}_V znaną z punktu 6.1.1 funkcją f_V . Odpowiednia formuła ma postać:

$$\forall a (\forall E \in \mathbf{E}_H (\forall A, A' \in E[\mathbf{A}_T] \quad ((a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \forall p \in P' (f_V(A', p) = f_V(A, p))) \rightarrow (A \models C(a) \leftrightarrow A' \models C(a)))))) \quad (6.16)$$

Jednakże parametry kontekstowe nie są częścią formalizmu metody SIM. Aby dowieść, że struktura kontekstowej bazy wiedzy wspiera pojęcie immanentności lokalnej, należy odnieść ją do właściwości tej struktury. Służy temu następujące twierdzenie:

Twierdzenie 6.2 (o immanentności lokalnej w SIM)

Dla każdego regularnego schematu kontekstowej bazy wiedzy H i dowolnych dwóch typów kontekstów $T, T' \in H[\mathbf{T}]$:

jeżeli $T' \in H[f_{agr}(T)] \wedge C \in Sig(T')$, to koncept C jest lokalnie immanentny w T względem $f_P(T)$. ■

Powyższe twierdzenie mówi, że każdy koncept, który występuje w sygnaturze jednego z typów kontekstów T' należących do dopuszczalnych miejsc agregacji danego typu kontekstu T , jest lokalnie immanentny w T względem parametrów przypisanych do T' .

Dowód:

Należy udowodnić następujące zdanie:

$$(T' \in H[f_{agr}(T)] \wedge C \in Sig(T')) \rightarrow \forall a (\forall E \in \mathbf{E}_H (\forall A, A' \in E[\mathbf{A}_T] \quad ((a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \forall p \in f_P(T') (f_V(A', p) = f_V(A, p))) \rightarrow (A \models C(a) \leftrightarrow A' \models C(a)))))) \quad (6.17)$$

Następnik pierwszej implikacji w tym zdaniu, to formuła (6.16) po podstawieniu $P' = f_P(T')$. Przeprowadzony zostanie dowód nie wprost, więc należy wykazać, że zaprzeczenie następnika przy założeniu, że poprzednik jest prawdziwy prowadzi do sprzeczności. Zaprzeczeniem następnika (6.17) jest formuła:

$$\exists a (\exists E \in \mathbf{E}_H (\exists A, A' \in E[\mathbf{A}_T] \quad ((a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \forall p \in f_P(T') (f_V(A', p) = f_V(A, p))) \wedge (A \models C(a) \wedge A' \not\models C(a) \vee A \not\models C(a) \wedge A' \models C(a)))))) \quad (6.18)$$

Ponieważ wystąpienia A i A' są zamienne, można będzie usunąć jedną gałąź alternatywy.

W tym miejscu dowód się rozgałęzia w zależności od tego, czy (1) istnieje taki typ kontekstu T'' , że $\exists T'' \in H[f_{agr}(T)] (T \preceq T'' \wedge T'' \preceq T' \wedge \forall p \in f_P(T'') f_V(A, p) = f_V(A', p))$, czy też (2) taki typ nie istnieje. Jeżeli prawdziwe jest stwierdzenie (1), to należy wrócić do (6.17), wstawić T'' zamiast T' i powtórzyć kolejne kroki dowodu. Jeżeli prawdziwe jest (2), to zgodnie z poprzednikiem formuły (6.17) i z (6.18) wystąpienia A i A' spełniają twierdzenie 6.1 ponieważ: $f_{inst}(A) \preceq T'$ i $f_{inst}(A') \preceq T'$ (zgodnie z poprzednikiem (6.17)) oraz $\forall p \in f_P(T') (f_V(A', p) = f_V(A, p))$ (zgodnie z (6.18)). Musi zatem istnieć takie wystąpienie $A'' \in E[\mathbf{A}_T]$, które agreguje A i A' . Formuła (6.18) może być zapisana:

$$\exists a (\exists E \in \mathbf{E}_H (\exists A, A' \in E[\mathbf{A}_T] \quad ((a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \exists A'' \in E[\mathbf{A}_T] (A \ll A'' \wedge A' \ll A'')) \wedge A \models C(a) \wedge A' \not\models C(a)))) \quad (6.19)$$

Po usunięciu kwantyfikatorów, czyli przy założeniu, że istnieją spełniające (6.19): schemat H , typ kontekstu $T \in H[\mathbf{T}]$, instancja $E \in \mathbf{E}_H$, wystąpienia kontekstów $A, A' \in E[\mathbf{A}_T]$ oraz $A'' \in E[\mathbf{A}_T]$, formuła ta przybiera postać:

$$a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge A \ll A'' \wedge A' \ll A'' \wedge A \models C(a) \wedge A' \not\models C(a) \quad (6.20)$$

Formuła ta nie może być prawdziwa, gdyż z punktu 2) definicji 5.9 i punktu 3) definicji 5.10 oraz z definicji 5.7 wynika, że $(A \models C(a) \wedge A \ll A'') \rightarrow A'' \models C(a)$ oraz $(A' \not\models C(a) \wedge A' \ll A'') \rightarrow A'' \not\models C(a)$, a to powoduje, że po odpowiednim podstawieniu jest ona równoważna formule:

$$a \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge A'' \models C(a) \wedge A'' \not\models C(a).$$

Zaprzeczenie następnika zdania (6.17) doprowadziło więc do sprzeczności.

QED

Immanentność nie dotyczy tylko konceptów. W równym stopniu dotyczy ona ról:

Definicja 6.7 (immanentność lokalna ról w SIM)

Rola R jest *immanentna dla typu kontekstu T względem podzbioru parametrów P'*, gdzie $T \in H[\mathbf{T}]$ i $P' \subset f_P(T)$, wtwg

$$\forall a, b (\forall E \in \mathbf{E}_H (\forall A, A' \in E[\mathbf{A}_T] ((a, b \in \mathcal{S}(A) \cap \mathcal{S}(A') \wedge \bar{f}_v(A)[P'] = \bar{f}_v(A')[P']) \rightarrow (A \models R(a, b) \leftrightarrow A' \models R(a, b)))). \quad \blacksquare$$

W przypadku atrybutów definicja jest jeszcze prostsza, gdyż zakłada się, że w każdym kontekście widoczne są wszystkie możliwe ich wartości (nie są one częścią sygnatury).

Z powyższych rozważań widać, że w ramach struktury regularnej kontekstowej bazy wiedzy immanentność danego konceptu, roli lub atrybutu jest wymuszana przez właściwości relacji dziedziczenia i agregacji. Pojęcie to nie dotyczy już jakichś założeń metafizycznych, które twórca musi rozważyć w czasie procesu konceptualizacji, lecz konkretnych decyzji projektowych, których wynikiem będzie taki lub inny układ elementów tworzonej bazy wiedzy.

6.2 Problemy modelowania

Celem tego podrozdziału jest omówienie problemów związanych z projektowaniem hierarchicznych baz wiedzy. Opisanie w poprzednim podrozdziale zasady tworzenia struktury oraz nakreślona wizja sposobu, w jaki powinien działać system zgodny z metodą SIM, pozwalają lepiej określić zadania projektantów. Otóż ich zadaniem jest:

- 1) Zdefiniowanie listy parametrów kontekstowych oraz określenie:
 - a) wartości dla niektórych z nich,
 - b) ograniczeń dziedzin dla pozostałych.
- 2) Dokonanie podziału bazy wiedzy na typy kontekstów i nałożenie na zbiór typów kontekstów relacji dziedziczenia.
- 3) Określenie dla każdego typu kontekstu miejsc agregacji jego wystąpień.
- 4) Zdefiniowanie zawartości typów kontekstów adekwatnej do ustaleń z punktów poprzednich.

Zaproponowana kolejność zadań nie musi być kolejnością chronologiczną. Prawdopodobnie wszystkie zadania należy inicjować jednocześnie, chociaż wydaje się, że sama zawartość typów powinna być definiowana najpóźniej, gdyż zmiana zbioru typów kontekstów lub relacji dziedziczenia po dokonaniu konceptualizacji i rozmieszczeniu aksjomatów może być kosztowna (może na przykład dojść do utraty koherencji bazy). Czynności z trzech pierwszych punktów są ze sobą ściśle związane, gdyż zdefiniowanie listy parametrów kontekstowych narzuca sposób, w jaki dokonany zostanie podział na konteksty.

Z technicznego punktu widzenia podział ontologii na konteksty pociąga za sobą ograniczenie możliwości opisywania zależności między konceptami zdefiniowanymi w równoległych, czyli należących do dwóch różnych ścieżek dziedziczenia, typach kontekstów. Takie koncepty nie mogą być od siebie bezpośrednio uzależniane: nie może istnieć między nimi subsumcja, nie mogą istnieć role, których dziedzina byłaby opisana konceptami z jednego, a zakres konceptami z drugiego obszaru wiedzy. Gdyby zdefiniowano w nich koncepty o takiej samej nazwie, nie należy tych konceptów ze sobą kojarzyć. W skrajnym przypadku można kontekstową bazę wiedzy utworzyć z dwóch niezależnych ontologii O_1 i O_2 w ten sposób, że:

$$\mathcal{K} = (\{T_{max}, T(O_1), T(O_2)\} \cup \{A_{max}, A(O_1), A(O_2)\}, T_{max}, A_{max}, \{(T(O_1), T_{max}), (T(O_2), T_{max})\}, \{(A(O_1), A_{max}), (A(O_2), A_{max})\}, \{(A_{max}, T_{max}), (A(O_1), T(O_1)), (A(O_2), T(O_2))\})$$

$$f_{cont}(T_{max}) = \{ \}$$

$$f_{cont}(A_{max}) = \{ \}$$

Takie powiązanie między kontekstami całkowicie je od siebie odizoluje — brak będzie jakiegokolwiek przepływu wniosków. Dopiero wstawienie do T_{max} aksjomatów definiujących wspólne pojęcia i zależności między nimi otworzy możliwość stworzenia zależności między kontekstami zdefiniowanymi w kontekstach dziedziczących. Zależności te będą jednakże wnioskowane, czyli będą miały charakter ograniczeń strukturalnych. T_{max} odgrywa w tym wypadku taką samą rolę, jak ontologia fundacjonistyczna (patrz rozdział 3.2) używana w celu zunifikowania podstawowych pojęć łączonych ontologii. Oczywiście łączenie dwóch heterogenicznych ontologii jest zadaniem bardzo trudnym, gdyż chodzi w nim o ujednoczenie dwóch różnych, często zupełnie do siebie nie przystających konceptualizacji. Uwaga inżynierów jest skupiona na wyszukiwaniu podobieństw, różnic oraz sposobów niwelowania tych różnic. W sytuacji, gdy podział na konteksty wspiera tworzenie jednej, homogenicznej w założeniu konceptualizacji, problem polega na czymś zupełnie innym. Chodzi o to, aby jak najlepiej wykorzystać oferowane przez metodę możliwości do wyrażenia zamierzonych znaczeń. Uwaga projektantów musi być tym razem skierowana na to, w jaki sposób zaprojektować konteksty i ich przeznaczenia.

W punktach tego podrozdziału, 6.2.1 i 6.2.2, przedstawione zostały różne podejścia do sposobu wykorzystania kontekstów. Zamieszczono wiele przykładów obrazujących sposoby wykorzystania możliwości metody SIM do modelowania różnych sytuacji, a także omówiono wpływ rozważań z poprzedniego podrozdziału na różne decyzje projektowe.

6.2.1 Modelowanie różnych punktów widzenia

Z semantycznego punktu widzenia, podczas projektowania podziału bazy wiedzy na konteksty projektanci muszą sobie odpowiedzieć na pytanie, w jakim celu dany kontekst jest tworzony. Odpowiedzi na to pytanie powinna udzielić lista parametrów kontekstowych. Formalne ich właściwości zostały opisane w punkcie 6.1.1. Z punktu widzenia wpływu na semantykę parametry te, nawiązując do rozważań przedstawionych w punkcie 4.1.1, powinny

lokalizować kontekst w przestrzeni o wymiarach takich jak zakres (ang. *partiality*), poziom szczegółowości (ang. *approximation*) i punkt widzenia (ang. *perspective*). Znaczenie dwóch pierwszych wymiarów nie budzi większych wątpliwości. Właściwości metody SIM powodują, że idąc w dół hierarchii dziedziczenia i agregacji, dziedzina umieszczonego w kontekstach opisu jest coraz mniejsza (sygnatura wystąpienia kontekstu jest również coraz mniejsza, czyli opis obejmuje coraz mniejszą liczbę osobników), a poziom szczegółowości coraz większy (sygnatura typu kontekstu jest coraz większa, czyli obejmuje coraz większą liczbę nazw konceptów i ról). Jeśli chodzi o wymiar określający punkt widzenia, problem jest bardziej skomplikowany. Najczęściej definicję punktu widzenia wiąże się z istnieniem agentów posiadających różną wiedzę. Zgodnie z [BBG2000], reprezentacja uwzględniająca punkty widzenia musi także brać pod uwagę „przestrzenno-czasowe, logiczne i kognitywne spojrzenie na stan rzeczy” (w oryginale *a spatio-temporal, logical, and cognitive point of view on a state of affairs*). W podrozdziale 4.1 przedstawiono opis wielu różnych sposobów ujęcia problemu kontekstowych punktów widzenia w ramach logiki. Według logiki zaimków wskazujących (patrz punkt 4.1.2) punkt widzenia zależy od tego, przez kogo, kiedy, w jakim miejscu i przez jaki świat możliwy jest określany. Jednakże głównym celem metody SIM nie jest odzwierciedlanie przekonań wielu różnych agentów, lecz wykorzystanie struktury kontekstowej do opisu jednej homogenicznie ujętej wiedzy. Jak to zostało powiedziane we wstępie do niniejszej rozprawy, opisywanie posiadanej wiedzy za pomocą różnych rzutów, operowanie małymi jej fragmentami w toku rozumowania, jest najzupełniej zgodne z intuicją i ze stanem wiedzy psychologicznej. Dzięki temu z jednej strony odpada pewna liczba problemów (na przykład konteksty nie muszą być powiązane z różnymi agentami), z drugiej strony twórcy ontologii kontekstowej muszą nauczyć się korzystania z kontekstów, muszą zacząć zdawać sobie sprawę, że każdy opis, który tworzą musi mieć jakieś umiejscowienie z strukturze kontekstów.

Metoda SIM posiada dwa mechanizmy pozwalające opisywać różne punkty widzenia. Pierwszy mechanizm związany jest z cechami relacji dziedziczenia. Konteksty, które znajdują się w równoległych ścieżkach dziedziczenia, mogą niezależnie rozwijać konceptualizację przejętą z kontekstu, który jest ich wspólnym przodkiem. W ten sposób tworzone są różne języki, które są od siebie uzależnione tylko poprzez wspólnego przodka. Im bardziej ten wspólny przodek jest oddalony wzdłuż ścieżki dziedziczenia, im mniej pojęć definiuje, tym bardziej niezależne są języki. Korzystając z tej właściwości relacji dziedziczenia, twórca ontologii może dokonać opisu tej samej sytuacji, tego samego fragmentu dziedziny, używając różnych aparatów pojęciowych. W przykładzie 5.1 z rozdziału 5.1 przedstawiono sposób, w jaki z dwóch różnych punktów widzenia, korzystając z dwóch różnych słowników, można opisać ten sam zbiór osobników — mieszkańców miasteczka Kontekstowice. Jedna z terminologii dotyczy chóru, druga spraw socjalnych. Są to obszary wiedzy, które mają ze sobą niewiele wspólnego. Dlatego projektant może bezpiecznie założyć, że w przyszłości nie zaistnieje potrzeba uzależniania między sobą konceptów zdefiniowanych w ramach tych obszarów. Jednocześnie w sposób jednoznaczny da się wydzielić grupę pojęć, która jest wspólna dla obu obszarów i opisuje świat na wyższym poziomie uogólnienia. Przykładem takiego konceptu jest *Kobieta*. Dzięki tej grupie pojęć możliwe jest stworzenie dróg przepływu pewnych wniosków, które wzajemnie wzbogacają wiedzę zawartą w tych kontekstach.

Drugi mechanizm jest związany z właściwościami relacji konkretyzacji. Relacja ta pozwala na istnienie wielu wystąpień kontekstu tego samego typu. Dzięki temu, wykorzystując ten sam aparat pojęciowy, można stworzyć wiele różnych opisów faktów. W szczególności można z różnych punktów widzenia opisywać ten sam fragment dziedziny. Możliwość ta jest przedstawiona w przykładzie 6.1.

Przykład 6.1 (konteksty — fakty z różnych punktów widzenia)

Przykładowa ontologia składa się z dwóch typów kontekstów (ilustracja tego przykładu znajduje się na rys. 6.2):

$$\mathcal{K} = (\{T_1, T_2\} \cup \{A_1, A_2, A_3\}, T_1, A_1, \{(T_2, T_1)\}, \{(A_2, A_1), (A_3, A_1)\}, \{(A_1, T_1), (A_2, T_2), (A_3, T_3)\})$$

$$f_{cont}(T_1) = \{Umie_reanimować \sqsubseteq T\}$$

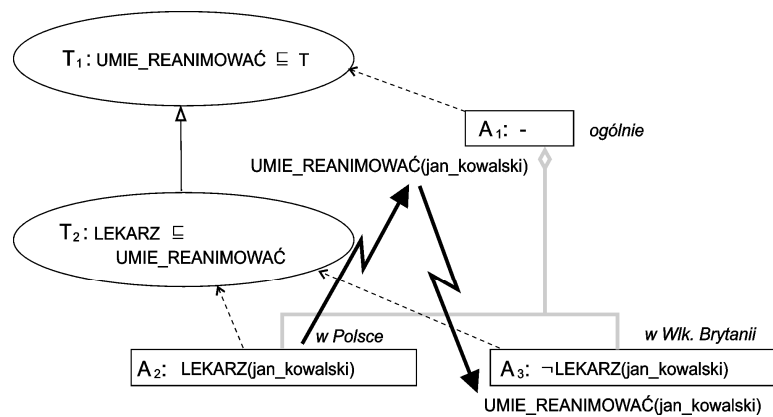
$$f_{cont}(T_2) = \{Lekarz \sqsubseteq Umie_reanimować\}$$

$$f_{cont}(A_1) = \{\}$$

$$f_{cont}(A_2) = \{Lekarz(jan_kowalski)\}$$

$$f_{cont}(A_3) = \{\neg Lekarz(jan_kowalski)\}$$

Typ T_2 zawiera definicję częściową konceptu *Lekarz*. Koncept ten dziedziczy od konceptu *Umie_reanimować* wprowadzonego w terminologii typu T_1 . Każde z dwóch wystąpień typu T_2 opisuje sytuację, w jakiej znajduje się osobnik *jan_kowalski*, z innego punktu widzenia: A_2 opisuje sytuację w Polsce, a A_3 w Wielkiej Brytanii. Człowiek, którego reprezentuje osobnik *jan_kowalski*, jest w Polsce lekarzem, czyli ukończył konieczne studia i odbył staż. Nie jest jednak lekarzem z punktu widzenia prawa brytyjskiego, gdyż nie dokonał koniecznej nostryfikacji dyplomu. Zdania *Lekarz(jan_kowalski)* i $\neg Lekarz(jan_kowalski)$ są zdaniami wzajemnie sprzecznymi, jednak zdefiniowanie konceptu *Lekarz* poniżej węzła agregacji wystąpień A_2 i A_3 pozwala uniknąć utraty spójności. Z kolei zdefiniowanie konceptu *Umie_reanimować* na poziomie węzła agregacji tych wystąpień sprawia, że staje się on dla typu T_2 immanentny. Dzięki temu możliwy jest przepływ wniosków odzwierciedlających fakt, że zarówno w Polsce, jak i w Wielkiej Brytanii, Jan Kowalski posiada wszelkie umiejętności, które powinien posiadać lekarz. Taką umiejętnością jest właśnie umiejętność przeprowadzenia reanimacji. ■



Rys. 6.2 Ilustracja kontekstowej bazy wiedzy zdefiniowanej w przykładzie 6.1.

Ten sam obiekt jest tu dwukrotnie opisany za pomocą tej samej terminologii. Dla każdego z nich przyjęto jednak inne tło, czyli inne wartości parametrów kontekstowych. Jeden opis jest dokonywany z punktu widzenia sytuacji, w jakiej obiekt ten znajduje się, będąc w Polsce, drugi — z punktu widzenia sytuacji, w jakiej obiekt ten znajduje się, przebywając w Wielkiej Brytanii. Oba opisy są wzajemnie spreczne, jednak dzięki odpowiedniemu rozłożeniu zawartości kontekstów i sposobu przepływu wniosków spójność ontologii została zachowana.

Jak widać, właściwości metody SIM pozwalają znacznie bardziej formalnie zdefiniować, czym są różne punkty widzenia. To, co w ramach innych opisanych w niniejszej pracy metod jest definiowane dość mgliście, tutaj przybiera kształt konkretnych wzorców projektowych.

6.2.2 Konteksty jako obiekty

Najbardziej naturalnym celem wyodrębnienia kontekstu jest zapewne utworzenie opisu jakiegoś obiektu należącego do modelowanej dziedziny. Opisy obiektów wydają się być właśnie tymi większymi cegiełkami wiedzy, o których mówił M. Minsky w cytacie przytoczonym we wstępie do rozdziału 5. Sposób opisanie obiektu, podobnie jak właściwości samego obiektu, które umysł ludzki wyodrębnia, by uporządkować postrzeganą rzeczywistość, są uzależnione jedynie od woli jego twórcy. Może mieć dowolną wielkość (czyli zakres), może być dowolnie uproszczony (czyli posiadać dowolny poziom szczegółowości), może ujmować dowolne, przydatne dla osiągnięcia jakiegoś celu cechy (czyli reprezentować jakiś punkt widzenia). Obiektowy opis problemów jest cechą charakterystyczną systemów tworzonych metodą relacyjną lub obiektową. W metodzie relacyjnej właściwości danego obiektu są opisane za pomocą składowych odpowiadającej temu obiektowi krotki. W metodzie obiektowej właściwości danego obiektu są reprezentowane wprost przez wartości atrybutów odpowiadającego temu obiektowi wystąpienia odpowiedniej klasy. W metodzie SIM typ kontekstu pozwala wyodrębnić fragment terminologii, który można uznać za definicję obiektów jakiejś klasy, a następnie w każdym z wystąpień tego kontekstu wyodrębnić taki fragment opisu świata, aby można go było uznać za opis właściwości obiektu tej klasy. Taki właśnie sposób wyodrębniania kontekstu jest przedstawiony w poniższym przykładzie.

Przykład 6.2 (kontekst jako obiekt)

Przyjmijmy, że zadaniem kontekstu jest opisanie grupy uczniów stanowiącej jednostkę organizacyjną w ramach jakiejś szkoły. W ramach szkół stopnia podstawowego i średniego byłaby to klasa, w ramach szkół wyższych grupy dziekańskie, specjalności, grupy lektoratowe itp. Tak zdefiniowany kontekst byłby częścią większej bazy wiedzy, której pełna zawartość nie jest w chwili obecnej przedmiotem naszego zainteresowania:

$$\begin{aligned}
 \mathcal{K} = & \left(\{T_{max}, \dots, T_1, \dots, T_2, \dots\} \cup \{A_{max}, \dots, A_1, \dots, A_2, A_3, \dots, A_7, \dots\}, T_{max}, A_{max}, \right. \\
 & \left. \{ \dots, (T_1, T_{max}), \dots, (T_2, T_1), \dots \}, \{ \dots, (A_1, A_{max}), \dots, (A_2, A_1), (A_3, A_1), \dots, (A_7, A_1), \dots \}^*, \right. \\
 & \left. \{ \dots, (A_{max}, T_{max}), \dots, (A_1, T_1), \dots, (A_2, T_2), (A_3, T_2), \dots, (A_7, T_2), \dots \} \right) \\
 f_{cont}(T_2) = & \{ Nauczyciel \sqsubseteq \neg Uczeń, \\
 & Opiekun \sqsubseteq Nauczyciel, \\
 & Jednostka \sqsubseteq \neg Nauczyciel \sqcap \neg Uczeń, \\
 & Jednostka \sqsubseteq \neg \text{ImaNazwę} \cdot T_{str}, \\
 & Gospodarz \sqsubseteq Uczeń, \\
 & Uczeń \sqcup Opiekun \sqsubseteq \neg \text{ImaNazwisko} \cdot T_{str} \} \\
 f_{cont}(A_2) = & \{ Jednostka(klasa1), maNazwę(klasa1, "Ia"), \\
 & Opiekun(nauczyciel1), maNazwisko(nauczyciel1, "Janina Kowalska"), \\
 & Gospodarz(uczeń1), maNazwisko(uczeń1, "Jerzy Nowak"), \\
 & maNazwisko(uczeń2, "Adam Mickiewicz"), maNazwisko(uczeń3, "Katarzyna Medycejska"), \\
 & \dots \} \\
 f_{cont}(A_3) = & \{ Jednostka(klasa2), maNazwę(klasa2, "Ib"), \\
 & Opiekun(nauczyciel2), maNazwisko(nauczyciel2, "Janina Nowacka"), \\
 & Gospodarz(uczeń28), maNazwisko(uczeń28, "Jerzy Kowal"), \\
 & maNazwisko(uczeń29, "Anna Karenina"), maNazwisko(uczeń30, "Stanisław Wokulski"), \\
 & \dots \} \\
 \dots \\
 f_{cont}(A_7) = & \{ Jednostka(klasa6), maNazwę(klasa6, "IIC"), \\
 & Opiekun(nauczyciel1), maNazwisko(nauczyciel1, "Janina Kowalska"), \\
 & Gospodarz(uczeń111), maNazwisko(uczeń111, "Juliusz Cezar"), \\
 & maNazwisko(uczeń2, "Wilhelm Zdobywca"), \\
 & \dots \}
 \end{aligned}$$

Typ kontekstu T_2 dziedziczy od typu T_1 , którego zawartość nie jest istotna (dlatego nie został rozpisany). Zakładamy jedynie, że T_1 nie definiuje żadnej z nazw definiowanych w T_2 . Podobnie jest z wystąpieniami: A_2 i A_3 są agregowane przez A_1 , którego zawartość nie jest istotna, ale zakładamy, że nie opisuje żadnego z osobników opisywanych w wystąpieniach agregowanych.

Typ kontekstu T_2 zawiera opis stanu rzeczy w ramach jednej jednostki organizacyjnej. Użyta została logika opisowa $\mathcal{ALCQ}(S)$, gdzie ostatni człon (S) oznacza dołączenie dziedziny konkretnej, której elementami są łańcuchy znakowe. Symbol \top_{str} oznacza koncept uniwersalny w ramach tej dziedziny. Opis stanu rzeczy jest adekwatny do przyjętego podziału na konteksty. Każde wystąpienie kontekstu opisuje jedną klasę (A_2 klasę Ia, A_3 klasę Ib).

Założmy, że naszą intencją jest, aby każde wystąpienie tego typu opisywało jedną klasę traktowaną jako grupa ludzi, aby obejmował zależności między tymi ludźmi i aby dotyczył określonego odcinka czasu, na przykład jednego semestru. Przypisując więc parametry kontekstowe typowi kontekstu T_2 , należy wziąć pod uwagę następujące czynniki:

- Zakres każdego opisu ma być ograniczony do jednej klasy, więc konieczny jest parametr mówiący, o którą klasę chodzi. Niech parametr ten nazywa się *klasa*. Ograniczony ma być też odcinek czasu, którego dany opis dotyczy, więc potrzebny będzie parametr o nazwie *semestr*.
- Poziom szczegółowości musi być taki sam dla wszystkich kontekstów, gdyż terminologia nie daje możliwości zróżnicowania opisu pod tym względem. Można jednak zastanowić się, czy nie znajdzie potrzeba bardziej szczegółowego opisu jakiegoś aspektu ujętego parametrami opisującymi zakres. Dlatego warto jednak dodać osobne parametry dotyczące szczegółowości i przyjąć, że mają pewną wartość początkową dla każdego kontekstu, na przykład: *jednostka organizacyjna* = „jedna klasa” i *zakres czasu* = „jeden semestr”.
- Punkt widzenia, podobnie jak poziom szczegółowości, jest taki sam dla wszystkich kontekstów. Terminologia pozwala opisać jedynie pewien rodzaj zależności między członkami grupy. Ale można sobie wyobrazić, że zależności te mogą być opisywane z jakiegoś innego punktu widzenia. Na przykład w czasie wycieczki szkolnej może być danej klasie przydzielony inny opiekun. Wobec tego także tu warto przyjąć parametr, założmy *typ zajęć*, który będzie miał narzuconą wartość początkową: „zajęcia dydaktyczne”.

Projektant przypisuje parametry kontekstowe typom kontekstów, a użytkownik, tworząc nowe wystąpienia, ustala ich wartości. W tym przykładzie parametry wystąpienia A_2 mogą mieć wartości:

- *klasa* = „Ia”,
- *semestr* = „zimowy 2009 – 2010”.

Pierwszy z parametrów został ujawniony (tego rodzaju operowanie parametrami nazywa się *push-pop* — patrz punkt 4.1.1) w ramach bazy wiedzy za pomocą atrybutu *maNazwę*. Tak jednak być nie musi: na przykład drugi z parametrów wciąż pozostaje niejawny. Decyzja o tym, które parametry kontekstowe powinny pozostać niejawne, a które powinny zostać ujawnione, należy do projektanta i jest uzależniona od przewidywanego sposobu używania danego kontekstu. Trzeba mieć też na uwadze, że użytkownik może dodać swoje własne, niejawne parametry kontekstowe, które mogą być zupełnie inne niż te, które przewidywali projektanci — w systemach relacyjnych także często zdarza się, że użytkownik wstawia do bazy danych krotki o znaczeniu zupełnie innym niż to, które zostało przewidziane w instrukcji obsługi. ■

Porównując projekt bazy wiedzy opisanej w powyższym przykładzie do odpowiadającego mu projektu niezmodularyzowanej bazy wiedzy, łatwo dostrzec zasadnicze różnice. Pierwszą, najbardziej rzucającą się w oczy, jest wyrażenie pewnej wiedzy nie w formułach logiki opisowej, lecz w samej strukturze kontekstów. W ontologii niezmodularyzowanej wyrażenie tej wiedzy wymagałoby zdefiniowania dodatkowych ról. Na przykład przyjęto, że nauczyciel o nazwisku Janina Kowalska jest opiekunem klasy Ia. Jest to wyrażone w ten sposób, że osobnik *nauczyciel1*, posiadający atrybut *maNazwisko* równy „Janina Kowalska”, jest wystąpieniem konceptu *Opiekun* w kontekście (T_2, A_2) , czyli w tym samym, w którym osobnik *klasa1*, posiadający atrybut *maNazwę* równy „Ia”, jest wystąpieniem konceptu *Jednostka*. W ontologii niezmodularyzowanej konieczne byłoby zdefiniowanie dodatkowej roli, która zostałaby użyta do powiązania osobnika *klasa1* z osobnikiem *nauczyciel1*: na przykład *jestOpiekunemKlasy(nauczyciel1, klasa1)*. W podobny sposób należałoby powiązać uczniów z ich klasą. Osiągnięta, dzięki takiemu sposobowi zamodelowania omawianego związku, oszczędność jest pozornie niewielka, jednak staje się istotna dla dużej ilości przechowywanych informacji. Z semantycznego punktu widzenia daje także pewne korzyści. Może się na przykład zdarzyć, że opiekun danej klasy się zmieni. W takiej sytuacji, chcąc tę zmianę odzwierciedlić w niezmodularyzowanej bazie wiedzy, należałoby dokonać reifikacji relacji stopnia wyższego niż drugi, zgodnie z zasadami podanymi w [NR2006]. W opisywanym przypadku musiałby zostać wprowadzony dodatkowy koncept, na przykład *PowiązanieWOkresie*, którego ekstensją byłby zbiór abstrakcyjnych osobników służących jedynie jako „zworniki” ról, oraz dodatkowa rola i atrybut, na przykład *zKlasą* i *obowiązujeWOkresie*, które pozwoliłyby przypisać takiemu „zwornikowi” jednostkę organizacyjną oraz okres obowiązywania. Dodatkowe aksjomaty i asercje musiałyby mieć następującą treść:

$$\begin{aligned} \text{PowiazanieWOkresie} \sqsubseteq & \neg \text{Opiekun} \sqcap \neg \text{Uczen} \sqcap \neg \text{Jednostka} \sqcap \\ & = \text{I} \text{obowiazujeWOkresie} . \top_{Str} \sqcap = \text{I} \text{zKlasa} . \text{Jednostka}, \\ \text{Uczen} \sqcup \text{Opiekun} \sqsubseteq & \exists \text{ jestOpiekunem} . \text{PowiazanieWOkresie}, \\ \text{PowiazanieWOkresie}(\text{pow1}), & \\ \text{jestOpiekunem}(\text{nauczyciel1}, & \text{pow1}), \\ \text{obowiazujeWOkresie}(\text{pow1}, & \text{„semestr zimowy 2009 - 2010”}), \\ \text{zKlasa}(\text{pow1}, \text{klasa1}) & \end{aligned}$$

W przykładzie 6.2 parametrom kontekstowym wystąpienia A_2 dotyczącym czasu zostały nadane następujące wartości:

- *semestr* = „zimowy 2009 – 2010”,
- *zakres czasu* = „jeden semestr”

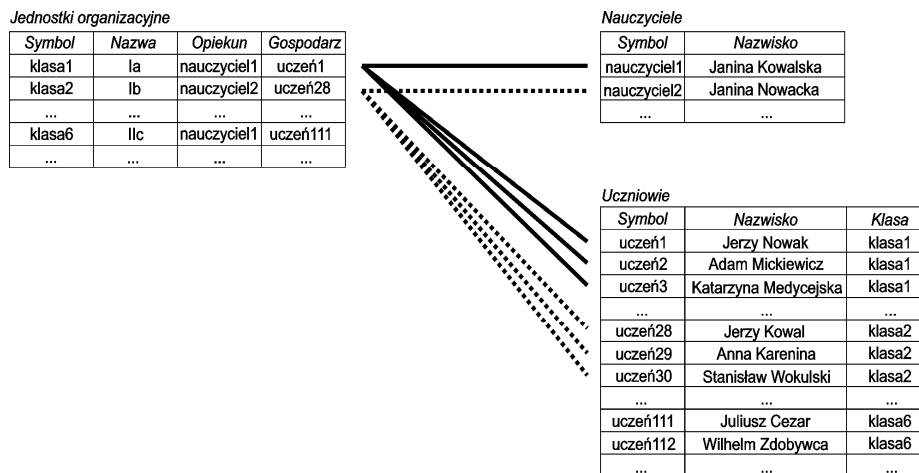
Tak więc po zmianie opiekuna można byłoby zmienić wartości tych parametrów dla wystąpienia reprezentującego klasę Ia (na przykład na *semestr*: „IX – X 2009” i *zakres czasu* = „część semestru”) i dodać do bazy nowe wystąpienie zawierające te same dane z wyjątkiem danych dotyczących opiekuna klasy i wartości parametru *semestr*, który określałby kolejny odcinek czasu. Zaletą takiego rozwiązania jest prostota, jednak nie należy zapominać, że wartość parametrów kontekstowych nie jest jawna z punktu widzenia zawartości bazy wiedzy. Dlatego, jeśli ujawnienie okresu obowiązywania danych zawartych w wystąpieniu kontekstu okaże się konieczne, należy to zrobić. Wystarczy jedynie dodać atrybut *obowiązujeWOkresie*:

$$f_{cont}(T_2) = f_{cont}(T_2) \cup \{ \text{Jednostka} \sqsubseteq = \text{I} \text{obowiazujeWOkresie} . \top_{Str} \} \quad (6.21)$$

$$f_{cont}(A_2) = f_{cont}(A_2) \cup \{ \text{obowiazujeWOkresie}(\text{klasa1}, \text{„IX 2009 - XI 2010”}) \} \quad (6.22)$$

W definicji tego atrybutu przyjęto, że obowiązuje on dla konceptu *Jednostka*, jednak nie jest to istotne, gdyż milcząco zakłada się, że dotyczy on całego wystąpienia kontekstu.

Oczywiście nic też nie stoi na przeszkodzie, aby opisywany problem rozwiązać w bazie kontekstowej podobnie, jak w bazie niezmodularyzowanej. Powyższa propozycja ma jedynie zilustrować możliwości, jakie daje wprowadzenie do bazy wiedzy dodatkowego poziomu opisu.



Rys. 6.3 Przedstawienie wiedzy zawartej w wystąpieniach kontekstów z przykładu 6.2 w postaci relacji.

Porównując projekt bazy wiedzy opisanej w przykładzie 6.2 do adekwatnego projektu, który mógłby być stworzony w ramach metody relacyjnej, można zauważyć, że istnieje pewna analogia. Rysunek 6.3 przedstawia przykładową instancję relacyjnej bazy danych o schemacie, który zostałaby prawdopodobnie przyjęty. Z pewnością utworzone zostałyby odrębne relacje dla jednostek organizacyjnych, nauczycieli i uczniów. Zostałyby one powiązane związkami, które przypisują klasie opiekuna, gospodarza i ucznia. Na rysunku 6.3 grubymi liniami zaznaczono „instancje” tych związków (czyli powiązania realizowane za pomocą kluczy obcych). Tworzą one grafy; każdy z grafów, łącznie z zawartością powiązanych krawędziami krotek, opisuje jeden obiekt: jednostkę organizacyjną rozumianą jako grupę ludzi w ramach określonego odcinka czasu (bieżący rok szkolny), opisując role tych ludzi. Kreską ciągłą oznaczony jest graf opisujący przynależność do klasy o symbolu *klasa1*, kreską przerywaną — do klasy o symbolu *klasa2*. Każdy taki obiekt jest więc informacyjnym odpowiednikiem jednego wystąpienia kontekstu. Analogia polega przede wszystkim na tym, że w obu metodach twórcy modelu dysponują możliwością dowolnego, zgodnego z ich potrzebami, grupowania informacji. Co więcej, sposób tego grupowania jest bardzo podobny, więc zgodny ze sposobem, w jaki współcześni inżynierowie modelują rzeczywistość. To, że w ramach ontologii, stworzonej zgodnie z metodą SIM, opis danego obiektu jest umieszczony w jednym kontekście, a w relacyjnej bazie danych muszą to być trzy relacje, nie jest istotną różnicą. Przede wszystkim podzielenie opisywanej informacji na trzy relacje jest czynnością arbitralną. Jest zupełnie możliwe utworzenie relacji, która w jednej krotce ujmie wszystkie dane w taki sposób, że zmieszczą się w jednej krotce¹⁹.

¹⁹ Podobne relacje tworzy się na potrzeby raportów. W opisywanym przykładzie przydatny mógłby być raport (do wywieszenia w hallu głównym danej szkoły), w którego każdym wierszu byłaby umieszczona jedna klasa, a następnie w kolejnych kolumnach nazwiska opiekuna i uczniów.

W praktyce takich relacji się nie tworzy, gdyż ich obsługa byłaby bardzo skomplikowana logicznie i trudna ze względów technicznych. Głównym problemem jest redundancja danych, która powoduje określone trudności podczas dodawania i usuwania krotek oraz modyfikowania ich zawartości. Unikaniu tych redundancji służy technika zwana normalizacją, która narzuca twórcom relacyjnych baz danych szereg reguł nakazujących bezstratny podział relacji nieznormalizowanych aż do uzyskania którejs z postaci normalnych. Choć taki zabieg ma swoje wady²⁰, to jednak zalety przeważają i przyjęło się, że bazy danych powinny być rozbite na wiele niedużych relacji, które spełniają kryteria reguł normalizacji. Niemożność przedstawienia w takiej strukturze bardziej rozbudowanych struktur grafowych rekompensuje wprowadzenie pojęcia związków — krotki różnych relacji powiązane kluczami obcymi („instancjami” związków) mogą już tworzyć grafy. Kontekst ontologiczny może zawierać dużo bardziej skomplikowany opis i dzięki temu nie ma potrzeby dzielić go na mniejsze fragmenty. Jednak, gdy mowa o unikaniu redundancji, trzeba zwrócić uwagę na to, że także w kontekstowej bazie wiedzy taki problem może wystąpić.

Przedstawione w przykładzie 6.2 konteksty są nieznormalizowane w takim samym sensie, w jakim nieznormalizowana może być relacja. Na przykład osobnik *nauczyciel1* pojawił się w dwóch wystąpieniach kontekstu: A_2 i A_7 . W obu wystąpieniach wpisano jego nazwisko, czyli wstawiono asercję $maNazwisko(nauczyciel1, „Janina Kowalska”)$. Pojawiła się w ten sposób redundancja danych i różne anomalie z nią związane. Sposób uniknięcia takiej redundancji jest podany w kolejnym przykładzie.

Przykład 6.3 (konteksty — unikanie redundancji)

Przedstawiona w tym przykładzie baza wiedzy jest rozwinięciem bazy z przykładu 6.2. Celem tego rozwinięcia jest uniknięcie redundancji związanej z powtarzaniem się asercji przypisującej wystąpieniu konceptu *Opiekun* wartość atrybutu *maNazwisko*. Zmiana polega na tym, że dla kontekstów typu T_1 zdefiniowano koncept *Człowiek* oraz atrybut *maNazwisko*. Umożliwiło to przypisanie nazwisk osobnikom reprezentującym nauczycieli w wystąpieniu agregującym A_1 . Wywołało to odpowiednie zmiany w kolejnych modułach bazy: T_2 , A_2 , A_3 i A_7 . W T_2 usunięty został aksjomat $Uczeń \sqsubseteq Opiekun \sqsubseteq =1maNazwisko.T_{Str}$ i zmieniono aksjomaty zaznaczone czcionką pogrubioną. W jego wystąpieniach usunięto asercje przypisujące nauczycielom nazwiska:

$$\begin{aligned} \mathcal{K} = & \left(\{T_{max}, \dots, T_1, \dots, T_2, \dots\} \cup \{A_{max}, \dots, A_1, \dots, A_2, A_3, \dots, A_7, \dots\}, T_{max}, A_{max}, \right. \\ & \left. \{ \dots, (T_1, T_{max}), \dots, (T_2, T_1), \dots \}^*, \{ \dots, (A_1, A_{max}), \dots, (A_2, A_1), (A_3, A_1), \dots, (A_7, A_1), \dots \}^*, \right. \\ & \left. \{ \dots, (A_{max}, T_{max}), \dots, (A_1, T_1), \dots, (A_2, T_2), (A_3, T_2), \dots, (A_7, T_2), \dots \} \right) \\ f_{cont}(T_1) = & \{ \dots \\ & \mathbf{Człowiek} \sqsubseteq =1maNazwisko.T_{Str}, \\ & \dots \} \\ f_{cont}(T_2) = & \{ \mathbf{Uczeń} \sqsubseteq \mathbf{Człowiek}, \\ & \mathbf{Nauczyciel} \sqsubseteq \mathbf{Człowiek} \sqcap \neg \mathbf{Uczeń}, \\ & \mathbf{Opiekun} \sqsubseteq \mathbf{Nauczyciel}, \\ & \mathbf{Jednostka} \sqsubseteq \neg \mathbf{Człowiek}, \\ & \mathbf{Jednostka} \sqsubseteq =1maNazwę.T_{Str}, \\ & \mathbf{Gospodarz} \sqsubseteq \mathbf{Uczeń} \} \\ f_{cont}(A_1) = & \{ \dots \\ & \mathbf{maNazwisko}(nauczyciel1, \mathbf{„Janina Kowalska”}), \\ & \mathbf{maNazwisko}(nauczyciel2, \mathbf{„Janina Nowacka”}), \\ & \dots \} \\ f_{cont}(A_2) = & \{ \mathbf{Jednostka}(klasa1, maNazwę(klasa1, \mathbf{„Ia”})), \\ & \mathbf{Opiekun}(nauczyciel1), \\ & \mathbf{Gospodarz}(uczeń1, maNazwisko(uczeń1, \mathbf{„Jerzy Nowak”})), \\ & \mathbf{maNazwisko}(uczeń2, \mathbf{„Adam Mickiewicz”}), \mathbf{maNazwisko}(uczeń3, \mathbf{„Katarzyna Medycejska”}), \\ & \dots \} \end{aligned}$$

²⁰ W [U1998] J. D. Ulmann pisze, że w początkowej fazie upowszechniania się modelu relacyjnego wśród jego pionierów miała miejsce dyskusja na temat stosowania relacji uniwersalnej, która zwalniałaby użytkownika z obowiązku poznania schematu bazy, przez co łatwiejsze byłoby tworzenie zapytań.

$$f_{com}(A_3) = \{ Jednostka(klasa2), maNazwę(klasa2, "Ib"), \\ Opiekun(nauczyciel2), \\ Gospodarz(uczeń28), maNazwisko(uczeń28, "Jerzy Kowal"), \\ maNazwisko(uczeń29, "Anna Karenina"), maNazwisko(uczeń30, "Stanisław Wokulski"), \\ \dots \}$$

$$\dots$$

$$f_{com}(A_7) = \{ Jednostka(klasa6), maNazwę(klasa6, "IIC"), \\ Opiekun(nauczyciel1), \\ Gospodarz(uczeń111), maNazwisko(uczeń111, "Juliusz Cezar"), \\ maNazwisko(uczeń2, "Wilhelm Zdobywca"), \\ \dots \}$$

Dzięki zasadom zachowania modelu z definicji 5.10 wartość atrybutu *maNazwisko* stała się cechą immanentną i jest dystrybuowana do wszystkich wystąpień agregowanych. Wystarczy więc jedna asercja w wystąpieniu A_1 , aby nazwisko nauczyciela było znane we wszystkich tych wystąpieniach, w których za pomocą asercji *Opiekun(x)* (lub jakiejś innej) osobnik, który tego nauczyciela reprezentuje, został wprowadzony do sygnatury.

Zdefiniowanie atrybutu *maNazwisko* na poziomie wystąpienia agregującego otworzyło również możliwość przepływu informacji w zakresie nazwisk uczniów. Z tego powodu byłoby bardziej bezpieczne umieszczenie asercji określających ich nazwiska również w wystąpieniu A_1 . Gdyby bowiem zdarzyło się, że dany uczeń jest zapisany do dwóch klas, to mogłoby zaistnieć niebezpieczeństwo przypisania mu dwóch różnych nazwisk, czyli wywołania niespójności bazy. Z drugiej strony, przy ograniczeniu mówiącym, że uczeń może być wpisany do jednej tylko klasy, taka niespójność może być pożytecznym sygnałem informującym, że popełniono błąd łamiący to ograniczenie. ■

Jak widać w podanym przykładzie, dzięki odpowiedniemu w stosunku do węzłów agregacji rozmieszczeniu aksjomatów i asercji możliwe jest takie wykorzystanie właściwości metody SIM, aby uzyskać skutek podobny do normalizacji w modelu relacyjnym.

Kolejny przykład przedstawia sposób, w jaki można właściwości metody SIM wykorzystać do opisanego danego obiektu z różnych punktów widzenia. W tym przypadku chodzi o opis klasy z punktu widzenia różnych nauczanych przedmiotów.

Przykład 6.4 (konteksty — obiekty z różnych punktów widzenia)

W tym przykładzie ponownie rozwijana jest baza wiedzy z poprzednich przykładów. Tym razem celem jest wprowadzenie bardziej szczegółowego poziomu opisu klasy z punktu widzenia różnych nauczanych przedmiotów. Do bazy wiedzy dodany został typ T_3 dziedziczący (być może tranzytywnie) od typu T_2 , który definiuje koncepty *NauczycielPrzedmiotu*, *Uzdolniony*, *Zagrożony* oraz rolę *maOcenę*. Pierwszy z konceptów pozwala przydzielić każdemu przedmiotowi innego nauczyciela. Jednocześnie, dzięki odpowiedniemu ustanowieniu subsumcji konceptów, każdy nauczyciel przedmiotu będzie należał do ekstensji konceptu *Nauczyciel* w wystąpieniach konceptu typu T_2 . Zadaniem konceptów *Uzdolniony* i *Zagrożony* jest wyodrębnienie uczniów, którzy mają zdolności do danego przedmiotu lub mają problemy z jego zaliczeniem. Rola *maOcenę* pozwala przypisać poszczególnym uczniom oceny jako atrybuty ze zbioru liczb naturalnych z przedziału [1, 6]. Dlatego konieczne było zwiększenie ekspresywności logiki opisowej do $\mathcal{ALCQ}(NS)$, gdzie N oznacza dziedzinę liczb naturalnych.

Dodanie nowego typu kontekstu wiąże się z koniecznością przydzielenia mu zbioru parametrów kontekstowych. Oczywiście, zgodnie z rozważaniami z punktu 6.1.1, przejmuje on wszystkie parametry od typu dziedzicznego. Do dotychczasowego parametru opisującego punkt widzenia (*typ zajęć*) musi być dodany nowy parametr: *przedmiot*.

Do bazy wiedzy dodano też dwa przykładowe wystąpienia A_{22} i A_{23} . Ich węzłem agregacji jest wystąpienie A_2 , opisujące klasę Ia.

$$\begin{aligned}
\mathcal{K} = & \left(\{T_{max}, \dots, T_1, \dots, T_2, \dots, T_3, \dots\} \cup \{A_{max}, \dots, A_1, \dots, A_2, A_3, \dots, A_7, \dots, A_{22}, A_{23}, \dots\}, T_{max}, A_{max}, \right. \\
& \left. \{ \dots, (T_1, T_{max}), \dots, (T_2, T_1), \dots, (T_3, T_2), \dots \}^*, \right. \\
& \left. \{ \dots, (A_1, A_{max}), \dots, (A_2, A_1), (A_3, A_1), \dots, (A_7, A_1), \dots, (A_{22}, A_2), (A_{23}, A_2), \dots \}^*, \right. \\
& \left. \{ \dots, (A_{max}, T_{max}), \dots, (A_1, T_1), \dots, (A_2, T_2), (A_3, T_2), \dots, (A_7, T_2), \dots, (A_{22}, T_3), (A_{23}, T_3), \dots \} \right) \\
f_{cont}(T_1) = & \{ \dots \\
& \text{Człowiek} \sqsubseteq \text{lmaNazwisko.T}_{Str}, \\
& \dots \} \\
f_{cont}(T_2) = & \{ \text{Uczeń} \sqsubseteq \text{Człowiek}, \\
& \text{Nauczyciel} \sqsubseteq \text{Człowiek} \sqcap \neg \text{Uczeń}, \\
& \text{Opiekun} \sqsubseteq \text{Nauczyciel}, \\
& \text{Jednostka} \sqsubseteq \neg \text{Człowiek}, \\
& \text{Jednostka} \sqsubseteq \text{lmaNazwę.T}_{Str}, \\
& \text{Gospodarz} \sqsubseteq \text{Uczeń}, \\
& \text{Uczeń} \sqcup \text{Opiekun} \sqsubseteq \text{lmaNazwisko.T}_{Str} \} \\
f_{cont}(T_3) = & \{ \text{Uczeń} \sqsubseteq \exists \text{maOcena} \geq_1 \wedge \leq_6, \\
& \text{NauczycielPrzedmiotu} \sqsubseteq \text{Nauczyciel}, \\
& \text{Uzdolniony} \sqsubseteq \text{Uczeń}, \\
& \text{Zagrożony} \sqsubseteq \text{Uczeń} \sqcap \neg \text{Uzdolniony} \} \\
f_{cont}(A_1) = & \{ \dots \\
& \text{maNazwisko}(\text{nauczyciel1}, \text{"Janina Kowalska"}), \\
& \text{maNazwisko}(\text{nauczyciel2}, \text{"Janina Nowacka"}), \\
& \dots \} \\
f_{cont}(A_2) = & \{ \text{Jednostka}(\text{klasa1}, \text{maNazwę}(\text{klasa1}, \text{"Ia"}), \\
& \text{Opiekun}(\text{nauczyciel1}), \\
& \text{Gospodarz}(\text{uczeń1}, \text{maNazwisko}(\text{uczeń1}, \text{"Jerzy Nowak"}), \\
& \text{maNazwisko}(\text{uczeń2}, \text{"Adam Mickiewicz"}), \text{maNazwisko}(\text{uczeń3}, \text{"Katarzyna Medycejska"}), \\
& \dots \} \\
f_{cont}(A_3) = & \{ \text{Jednostka}(\text{klasa2}, \text{maNazwę}(\text{klasa2}, \text{"Ib"}), \\
& \text{Opiekun}(\text{nauczyciel2}), \\
& \text{Gospodarz}(\text{uczeń28}, \text{maNazwisko}(\text{uczeń28}, \text{"Jerzy Kowal"}), \\
& \text{maNazwisko}(\text{uczeń29}, \text{"Anna Karenina"}), \text{maNazwisko}(\text{uczeń30}, \text{"Stanisław Wokulski"}), \\
& \dots \} \\
& \dots \\
f_{cont}(A_7) = & \{ \text{Jednostka}(\text{klasa6}, \text{maNazwę}(\text{klasa6}, \text{"IIC"}), \\
& \text{Opiekun}(\text{nauczyciel1}), \\
& \text{Gospodarz}(\text{uczeń111}, \text{maNazwisko}(\text{uczeń111}, \text{"Juliusz Cezar"}), \\
& \text{maNazwisko}(\text{uczeń2}, \text{"Wilhelm Zdobycwa"}), \\
& \dots \} \\
& \dots \\
f_{cont}(A_{22}) = & \{ \text{NauczycielPrzedmiotu}(\text{nauczyciel11}), \\
& \text{Uzdolniony}(\text{uczeń18}), \\
& \text{Zagrożony}(\text{uczeń22}), \\
& \text{Zagrożony}(\text{uczeń24}), \\
& \text{maOcena}(\text{uczeń20}, 6), \\
& \dots \} \\
f_{cont}(A_{23}) = & \{ \text{NauczycielPrzedmiotu}(\text{nauczyciel18}), \\
& \text{Zagrożony}(\text{uczeń18}), \\
& \dots \} \\
& \dots
\end{aligned}$$

Każde z tych wystąpień opisuje tę klasę z punktu widzenia innego przedmiotu. Wartości parametrów kontekstowych, w zakresie parametrów dziedziczonych, nie różnią się od wartości określonych dla A_2 , czyli (przypomnijmy):

- *klasa* = „Ia”,
- *semestr* = „zimowy 2009 – 2010”,

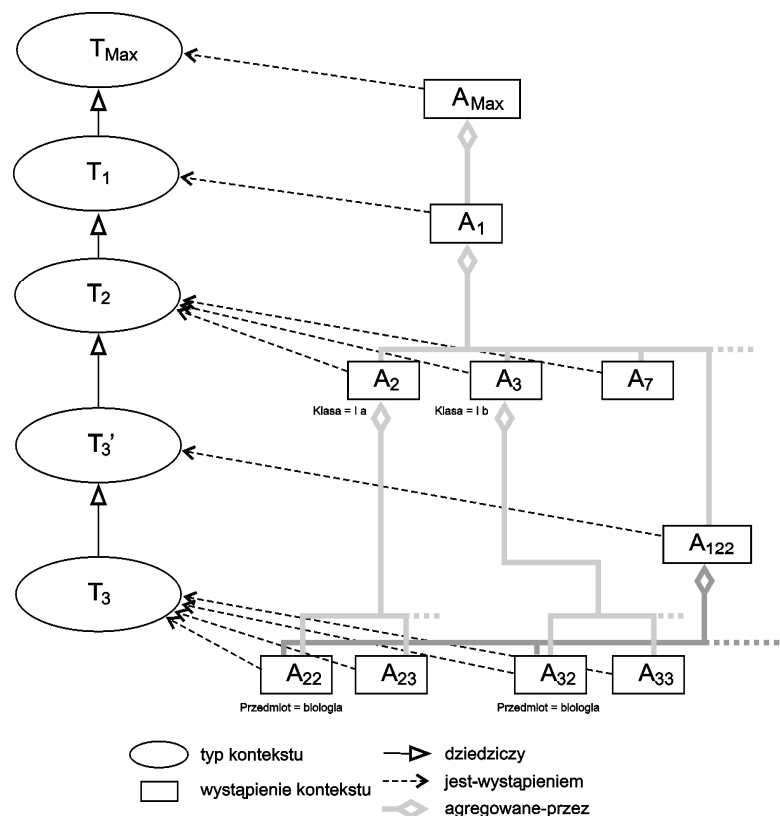
ale musi zostać określony parametr mówiący o tym, którego przedmiotu dotyczą, na przykład:

- *przedmiot* = „biologia”. ■

W powyższym przykładzie każdy z „obiektów” — wystąpień kontekstów reprezentujących daną klasę — jest „rozszczepiany” na szereg nowych wystąpień, które pozwalają opisać tę klasę z punktu widzenia każdego z nauczanych przedmiotów. Na przykładzie osobnika *uczeń18* widać, że w różnych wystąpieniach tego kontekstu ten sam osobnik może być przypisany do różnych, nawet wzajemnie rozłącznych konceptów. W węzle agregacji, którym dla wystąpień A_{22} i A_{23} jest wystąpienie A_2 , można uzyskać

informację zbiorczą dotyczącą wszystkich przedmiotów. Na przykład w każdym z wystąpień typu T_3 jest wpisany jedynie jeden osobnik będący wystąpieniem konceptu *NauczycielPrzedmiotu*, który reprezentuje nauczyciela prowadzącego dany przedmiot. W wystąpieniu agregującym wszystkie te osobniki będą należały do ekstensji konceptu *Nauczyciel*, dzięki czemu znani będą nauczyciele wszystkich przedmiotów nauczanych w danej klasie (na przykład w A_2 będą to wszyscy nauczający w klasie I a). Brak jednak byłoby informacji o tym, który nauczyciel naucza którego przedmiotu. Gdyby taka informacja była potrzebna, należałoby ujawnić niejawni dotąd parametr opisujący przedmiot nauczania, podobnie jak w wyrażeniach (6.21) i (6.22) został ujawniony parametr dotyczący okresu obowiązywania danych o klasie.

W analogiczny sposób, jak w przypadku informacji o przedmiotach nauczania, można sformułować opis klasy z różnych innych punktów widzenia: badań lekarskich, wywiadu środowiskowego, zainteresowań uczniów itp. Aby to zrealizować, należy zdefiniować nowy typ kontekstu dziedziczący od typu T_2 (i być może jeszcze innych typów znajdujących się w równoległych ścieżkach dziedziczenia), a w ramach tego typu terminologię odpowiednią do danego sposobu widzenia klasy. W pewnym sensie, w analogii do modelu obiektowego, jest to czynność podobna do nakładania na dany obiekt różnych interfejsów.



Rys. 6.4 Agregowanie wystąpień kontekstów w kilku węzłach agregacji. Wystąpienie A_{22} jest łącznie z wystąpieniem A_{23} agregowane przez wystąpienie A_2 , a z wystąpieniem A_{32} przez wystąpienie A_{122} .

Stwierdzenie, że kontekst można postrzegać jako obiekt, jest trafne, ale należy wziąć pod uwagę, że możliwości metody SIM idą o wiele dalej. Warto przeanalizować jeszcze jeden przykład. Tym razem jest on przedstawiony jedynie na rysunku 6.4. Rysunek ten pokazuje

strukturę kontekstów podobną do struktury zdefiniowanej w przykładzie 6.4. Różnica polega na tym, że dodano typ kontekstu T_3' oraz jego wystąpienie A_{122} . Typ T_3' jest opisany dokładnie taką samą terminologią, jak typ T_3 . Został dodany tylko i wyłącznie ze względu na to, że, zgodnie z definicją 5.3, relacja agregacji może być przeprowadzona jedynie między wystąpieniami kontekstów, których typy znajdują się w relacji dziedziczenia (relacja dziedziczenia jest przeciwzrotna). Wystąpienie A_{122} agreguje wszystkie wystąpienia (na przykład A_{22} , A_{32}), które opisują klasy z punktu widzenia nauczania biologii. Dzięki takiemu ujęciu nauczyciel biologii zyskuje całościowe spojrzenie na wszystkich swoich uczniów ze wszystkich klas.

6.3 Podsumowanie

W bieżącym rozdziale przedstawione zostały semantyczne cechy struktury kontekstowej bazy wiedzy. Cechy te są realizacją zasad kontekstowego wnioskowania przedstawionych w [BBG2000], a przytoczonych w punkcie 4.1.1. W rozdziale zostało wykazane, że struktura kontekstowej bazy wiedzy odpowiada treści zawartej w parametrach kontekstowych. Odpowiednie skojarzenie parametrów kontekstowych oraz ich wartości z elementami struktury pozwoliło na zdefiniowanie wielu pojęć, w tym pojęcia schematu i instancji bazy wiedzy. Odniesienie do metody OntoClean pozwoliło zdefiniować immanentność lokalną konceptów, ról i atrybutów. W ostatniej części rozdziału przedstawiono wiele przykładów omawiających problemy związane z projektowaniem kontekstowych baz wiedzy.

Największą siłą prezentowanej metody jest trafnie zdefiniowana semantyka elementów struktury oddających kontekstowy charakter wiedzy. Trafność tę potwierdza możliwość połączenia nomenklatury związanej z metodą SIM, zasad kontekstowego wnioskowania oraz metody OntoClean. Dzięki tej semantyce tworzone wewnątrz bazy moduły posiadają, w przeciwieństwie do metod przedstawionych w rozdziale 4, znaczenie z punktu widzenia modelowanego problemu. Relacje pomiędzy modułami w sposób jednoznaczny określają miejsce modułu wśród innych i ułatwiają projektantom orientowanie się w konstrukcji bazy. Ułatwia to zarówno zrozumienie jej zawartości, jak i podział obowiązków pomiędzy zespoły uczestniczące w pracach projektowych. Tym samym można uznać, że metoda rozwiązuje problem konstrukcyjny i organizacyjny.

Najważniejsze jednak jest to, że semantyka struktury kontekstowej bazy wiedzy nie opisuje metawarstwy narzędziowej, lecz jest semantyką integralnie związaną z semantyką tworzonego modelu. Elementy tej struktury współuczestniczą w opisanu dziedziny na równi z aksjomatami i asercjami, które są w nich zawarte. Tym samym metoda SIM rozwiązuje problem adekwatności kontekstów i problem projektowania kontekstowego.

7 Praktyczna realizacja ontologii zgodnej z metodą SIM

Metoda SIM została praktycznie wdrożona w ramach systemu wnioskującego KASEA. Wybór ten został podyktowany tym, że system KASEA, dzięki zastosowaniu wersji algorytmu strukturalnego, dobrze radzi sobie z obsługą bardzo dużych Aboxów. Dzięki temu spełnia wymagania wynikające z chęci upodobnienia sposobu korzystania z bazy wiedzy do sposobu korzystania z bazy danych. Fakt, że raczej słabo radzi sobie z dużymi Tboxami, nie był przeciwwskazaniem, gdyż terminologia w ramach metody SIM ma być dzielona między wiele typów kontekstów. Rozwiązanie takie było więc bardzo korzystne i dla systemu KASEA, i dla metody SIM. Podrozdział 7.1 opisuje sposób realizacji metody w ramach tego systemu.

System KASEA został wykorzystany do wnioskowania z ontologii PIPSDrugs, która pierwotnie wykonana w wersji niezmodularyzowanej, została przez autora niniejszej rozprawy podzielona na konteksty. Szczegóły dotyczące tej ontologii, a także uzyskane rezultaty są opisane w podrozdziale 7.2. Podrozdział 7.3 zawiera podsumowanie tego rozdziału.

7.1 System KaSeA

System KASEA to system wnioskowania z ontologii zapisanych w logice opisowej zaimplementowany w Katedrze Inżynierii Oprogramowania Wydziału Elektroniki i Telekomunikacji Politechniki Gdańskiej. Decyzję o implementacji systemu podjęto głównie w związku z uczestnictwem Wydziału w międzynarodowym projekcie PIPS (ang. *Personal Information Platform for Life and Health Services*), finansowanym w ramach 6. Programu Ramowego Unii Europejskiej (<http://www.pips.eu.org/>). Projekt ten bardzo intensywnie wykorzystywał ontologie do wymiany informacji pomiędzy komponentami tworzonej platformy oraz do pozyskiwania wiedzy spoza platformy. Niektóre informacje na temat prac prowadzonych w ramach tego projektu można znaleźć w [DGMS2006], [G2W2Z2006], [G2W2Z2007].

Zasadniczą motywacją dla utworzenia systemu KASEA były zidentyfikowane przez uczestników projektu niedostatki wydajności istniejących systemów wnioskujących dla baz wiedzy zawierających dużą liczbę osobników. Specyfika projektu PIPS, który zakresem swego zainteresowania obejmował koncepty o potencjalnie bardzo dużej liczbie wystąpień, takie jak schorzenia, składniki diety czy składniki leków, sprawiła, że konieczne stało się utworzenie mechanizmu wnioskującego działającego na odmiennym zasadzie.

Językiem porozumiewania się z systemem KASEA, obejmującym zarówno wprowadzanie aksjomatów i asercji, jak i zadawanie zapytań jest język DIGUT (http://km.pg.gda.pl/km/digut/1.3/DIGUT_Interface_1.3.pdf), który jest rozwinięciem języka DIG 1.1 (<http://dig.sourceforge.net/>). Wybór języka DIG był podyktowany tym, że był on w momencie dokonywania tego wyboru najlepiej rozwiniętym i jednym z najbardziej rozpowszechnionych standardów zadawania zapytań do baz wiedzy, który odzworowywał podstawowe problemy wnioskowania z ontologii wyrażonych formalizmem logiki opisowej. Inne języki, takie jak nRQL (<http://www.racer-systems.com/products/racerpro/manual.phtml>), język specjalnie zaprojektowany do komunikacji z system wnioskującym Racer Pro, lub SPARQL (<http://www.w3.org/TR/rdf-sparql-query/>), szeroko rozpowszechniony język zapytań do baz danych wyrażonych w standardzie RDF, nie oferowały takich możliwości.

Metodą wnioskowania wykorzystaną w Kasei była metoda kartograficzna. Metodę tę przedstawiono już w punkcie 5.3.3. Dla przypomnienia warto jednak powtórzyć, że najważniejszymi pojęciami tej metody były mapy, przedstawiające zależności pomiędzy

konceptami w terminologii poprzez wskazanie spełnialnych konceptów należących do pewnej dekompozycji, oraz rozmieszczenia, przyporządkowujące osobniki do konceptów (regionów) zawartych w mapie.

Wykorzystanie metody kartograficznej pozwoliło na zastosowanie nieco innego paradygmatu przeprowadzania wnioskowania. Wnioskowanie w Kasei, w przeciwieństwie do większości istniejących systemów dla logiki opisowej, prowadzone jest w przód. Oznacza to, że za każdym razem, kiedy użytkownik przeprowadza aktualizację bazy wiedzy (poprzez dodanie nowej asercji), system wnioskujący podejmuje pracę i poprzez utworzenie nowej wersji rozmieszczenia produkuje i zapamiętuje całą klasę wniosków wynikających z nowo wprowadzonej informacji. Wnioski te są natychmiast dostępne dla użytkowników, co znacznie skraca czas wykonania zapytania.

KASEA zawiera wiele optymalizacji usprawniających jej pracę. Przede wszystkim system ten wykorzystuje relacyjną bazę danych do reprezentowania i przechowywania rozmieszczeń. Wykorzystanie bazy danych pozwala na stosunkowo łatwe wskazanie klas podobnych osobników (tj. osobników o takich samych wartościach funkcji rozmieszczenia), dzięki czemu informacje na ich temat mogą być przechowywane tylko raz i bez redundancji. Samo rozmieszczenie konstruowane jest w sposób inkrementacyjny, tak że jego reprezentacja jest jedynie lokalnie aktualizowana w miarę przetwarzania nowych asercji.

Zarówno mapy, jak i rozmieszczenia generowane przez Kaseę są wewnętrznie reprezentowane za pomocą tzw. *deskryptorów binarnych*²¹. Warunkiem wstępnym dla utworzenia deskryptorów jest poindeksowanie mapy konceptów M za pomocą zbioru J (poindeksowana może być również zastosowana dekompozycja \mathbf{D} , co prowadzi do analogicznych rezultatów, choć w tym przypadku trzeba uwzględnić pewne dodatkowe założenia), tj. przyjęcia, że $M = \{C_i \mid i \in J\}$. Deskryptorem konceptu D nazwiemy wówczas ciąg $b(D) = (b_i)_{i \in J}$, taki że $b_i \in \{0, 1\}$ oraz $b_i = 1$ wtwg $KB \models C_i \sqsubseteq D$ ²², przy czym stosuje się tu ogólną zasadę mówiącą, że deskryptory przypisywane są jedynie konceptom zawartym w ekspresywności przyjętej dekompozycji (w takim ujęciu b może być potraktowane jako funkcja $Ex(\mathbf{D}) \rightarrow \{0,1\}^{|M|}$). Z kolei deskryptor osobnika to ciąg $b(a) = (b_i)_{i \in J}$, taki że $b_i \in \{0, 1\}$ oraz $b_i = 1$ wtwg $KB \models \neg C_i(a)$, lub, równoważnie, kiedy $C_i \in r(a)$.

W Kasei wykorzystywano prostą algebrę deskryptorów, będącą wariantem algebry Boole'a dla ciągów binarnych, której działania pozwalały na rozwiązanie podstawowych problemów wnioskowania z terminologii oraz terminologii i opisu świata (np. podrzędność konceptów C i D mogła być stwierdzona poprzez zbadanie, czy $b(C) \wedge b(D) = b(C)$; patrz [W2007]). Same deskryptory przechowywane były natomiast w bazie danych w oryginalnej postaci, jako ciągi binarne, przy czym jednemu elementowi ciągu odpowiadał jeden bit.

W ramach prac rozwojowych Kasei dokonano jej adaptacji tak, aby obsługiwała ona metodę SIM. Adaptacja ta odbyła się zgodnie ze schematem opisanym w punkcie 5.3.3.1, czyli obejmowała migrację od pojęć mapy i rozmieszczenia do mapy kontekstu i rozmieszczenia dla kontekstu. Oznaczało to, że korekcie musiało ulec także pojęcie deskryptora: w nowej wersji deskryptor był przypisywany konceptom z każdego typu kontekstu i osobnikom z każdego wystąpienia kontekstu. Mówiąc ściślej, indeksowanie objęło mapy wszystkich kontekstów $M(\kappa) = \{C_i\}_{i \in J_\kappa}$, zaś deskryptorem konceptu D w kontekście κ nazwano ciąg $b(D) = (b_i)_{i \in J_\kappa}$ taki że $b_i \in \{0, 1\}$ oraz $b_i = 1$ wtwg $\mathcal{K} \models_\kappa C_i \sqsubseteq D$. Z kolei deskryptorem

²¹ W poprzednich wersjach Kasei stosowana była nazwa *sygnatura binarna*, którą to konwencję zarzucono ze względu na chęć jasnego odróżnienia tego pojęcia od ważnego pojęcia sygnatury rozumianej jako zbiór nazw.

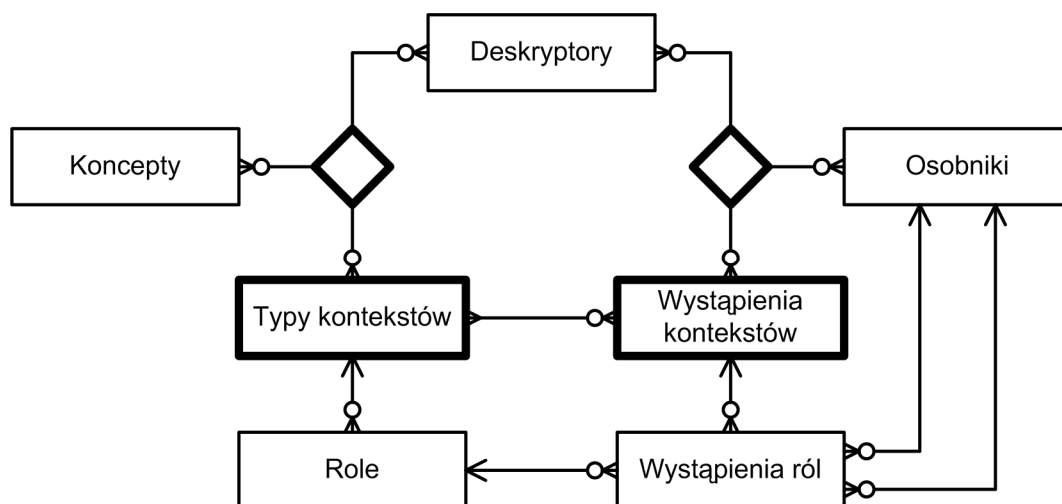
²² Zachodzenie tego ostatniego faktu ($KB \models C_i \sqsubseteq D$) może być stwierdzone przez sprawdzenie, czy $C_i \subseteq f_D(D)$.

osobnika a w kontekście κ nazwano ciąg $b(a) = (b_i)_{i \in J_\kappa}$ taki że $b_i \in \{0, 1\}$ i $b_i = 1$ wtwg $C_i \in r(\kappa)(a)$.

Same algorytmy zostały dostosowane poprzez dodanie do nich mechanizmów iteracyjnego przenoszenia wniosków, analogicznie jak opisano to w 5.3.3.1. Przenoszenie wniosków w nowej wersji odbywa się poprzez aktualizację odpowiednich deskryptorów (jednocześnie z wyznaczaniem nowego rozmieszczenia).

Aktualizacja systemu pociągnęła za sobą konieczność zmiany struktury bazy danych. Na rysunku 7.1 przedstawiono model danych E-R przeznaczony dla kontekstowych baz wiedzy, przy czym nowo wprowadzone elementy zostały wyróżnione pogrubieniem. Jak widać, każdy koncept i osobnik został wyposażony w zestaw deskryptorów (zamiast pojedynczego deskryptora), przy czym deskryptor ten przyporządkowany jest odpowiednio typowi lub wystąpieniu kontekstu.

Tak zaadaptowany system został wykorzystany w projekcie PIPS (scenariusz *Smart Cabinet*) do wnioskowania z kontekstowej ontologii PIPSDrugs opisanej w kolejnym podrozdziale.



Rys. 7.1 Model danych systemu KASEA w wersji zaadaptowanej do obsługi baz wiedzy SIM.

7.2 Ontologia PIPSDrugs

Metoda SIM została stworzona z myślą o projektowaniu nowych ontologii, o dokonywaniu konceptualizacji, które z założenia będą uwzględniały kontekstowy charakter wiedzy. Nie istnieje żadna gwarancja, że dana ontologia, która została utworzona przy wykorzystaniu innych metod projektowych, da się przekształcić na ontologię o proponowanej w tej pracy strukturze hierarchiczno-kontekstowej. Podjęcie więc decyzji o dokonaniu takiego podziału jest obarczone dużym ryzykiem. Jednakże próba ta ma dużą wartość badawczą. Z jednej strony bowiem interesujące jest w jaki sposób metoda SIM wypadnie w konfrontacji z ontologią, która została zaprojektowana w celu rozwiązania rzeczywistych problemów. Z drugiej strony ważne jest, aby sprawdzić, czy wiedza sformułowana w poprawnie zaprojektowanej ontologii niezmodularyzowanej da się wyrazić w sposób kontekstowy, czy wprowadzona kontekstowość da jakieś korzyści, czy raczej spowoduje jakieś problemy. Innymi słowy mówiąc, czy założenie o kontekstowym charakterze ludzkiej wiedzy jest

śluszne. Przyjmując takie założenie, należy się spodziewać, że konteksty istnieją niejawnie w każdej bardziej złożonej ontologii, należy je jedynie wykryć i ujawnić.

Niniejszy podrozdział przedstawia opis kontekstualizacji rzeczywistej ontologii dokonany przez autora tej rozprawy podczas prac nad projektem PIPS (część z przedstawionych tu wyników opisano również w [GWW2009e]). W punkcie 7.2.1 przedstawiona jest ogólna charakterystyka ontologii, zanim jeszcze została ona podzielona na konteksty. Punkt ten zawiera również krótki opis metody oceny stopnia złożoności ontologii, która została zastosowana podczas opisu skutków przeprowadzonego podziału. W punkcie 7.2.2 opisano wymagania, jakie powinna spełnić ontologia po podziale, oraz sposób, w jaki podział został dokonany. Punkt 7.2.3 przedstawia ocenę uzyskanych wyników.

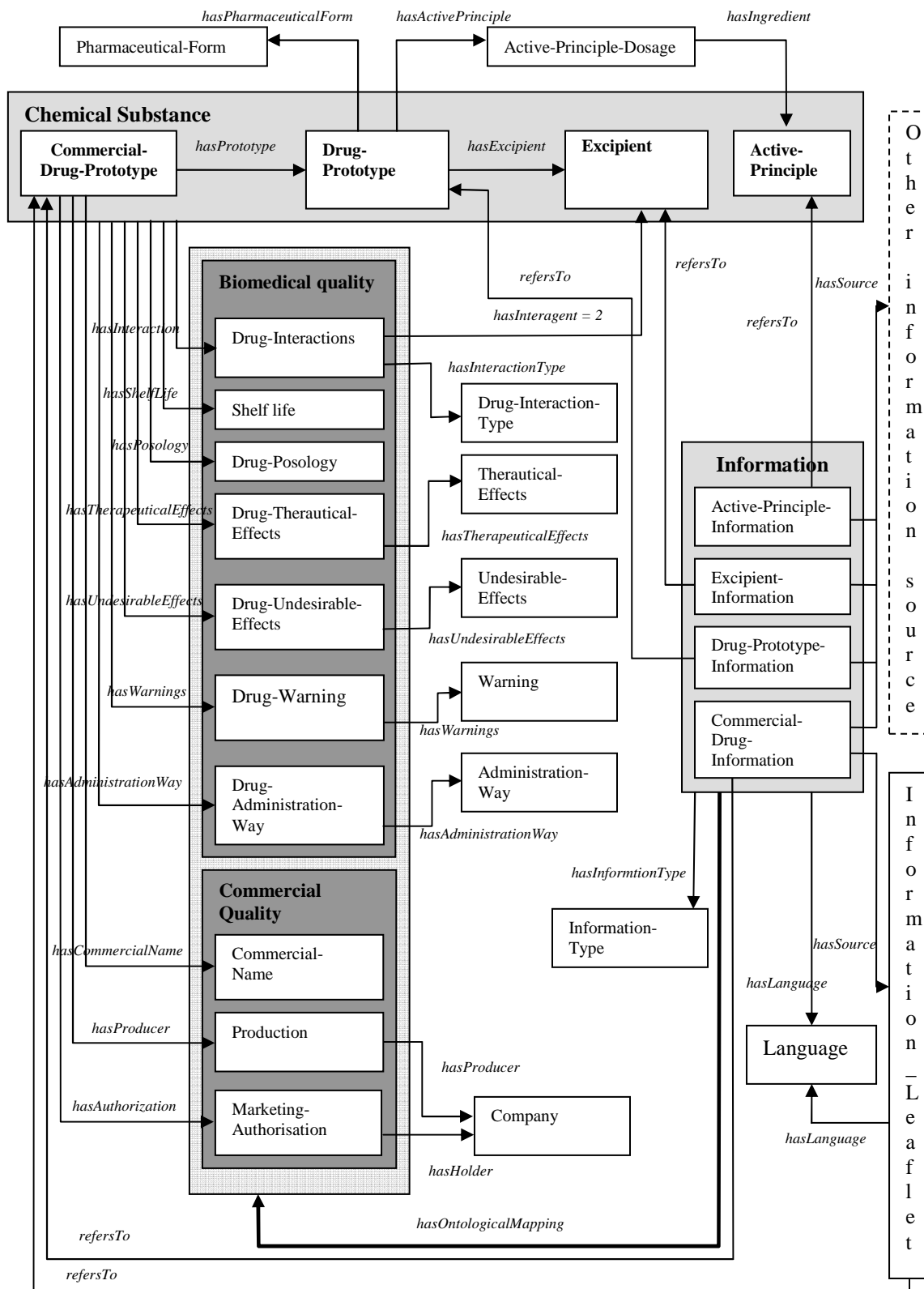
7.2.1 Charakterystyka ontologii PIPSDrugs

Ontologia PIPSDrugs powstała w ramach scenariusza *Smart Cabinet*, realizowanego w trakcie prac nad projektem PIPS. Sam projekt miał na celu stworzenie systemu, który składał się z wielu komponentów [DGMS2006]. Komponenty te realizowały kilka głównych scenariuszy, z których dwa najważniejsze były związane z opieką nad osobami przewlekle chorymi i z promowaniem zdrowych sposobów odżywiania się. Scenariusze główne były podzielone na scenariusze pomocnicze. *Smart Cabinet* był scenariuszem pomocniczym związanym z opieką nad chorymi i obejmował szereg usług skoncentrowanych na decyzjach związanych z przepisywaniem i przyjmowaniem leków. Podstawowym zasobem informacji związanym z tym scenariuszem była właśnie ontologia PIPSDrugs. Została ona stworzona przez zespół HSR (wł. *Fondazione Centro San Raffaele del Monte Tabor*, <http://www.sanraffaele.org/>), partnera uczestniczącego w projekcie. Zespół ten tworzył także główne oprogramowanie dla scenariusza.

Ontologia PIPSDrugs została wyrażona w formalizmie logiki opisowej *SHOIN(D)* w języku OWL-DL. Zawiera opis 214 konceptów (134 atomowych i 80 złożonych) oraz 151 właściwości. Jest to więc ontologia średniej wielkości. Jednakże posiada ona cechy powodujące, że nie jest łatwa do przetwarzania. Obejmuje bowiem wiele grup tematycznych związanych z lekami, takich jak skład chemiczny substancji chemicznych tworzących lek, informacje o właściwościach komercyjnych, właściwościach biomedycznych, a także dane dotyczące informatorów publikowanych na temat leków. Każda z takich grup tematycznych zawiera zbiór pojęć wyspecjalizowanych, specyficznych jedynie dla niej. Widać to na uproszczonym schemacie przedstawionym na rys. 7.2. Pojęcia te są w ramach swoich grup wzajemnie powiązane, lecz brak jest powiązań między pojęciami specyficznymi dla różnych grup. Ponadto istnieje pewna liczba pojęć wspólnych, które są powiązane z pojęciami wyspecjalizowanymi. Dodatkowo ontologia PIPSDrugs jest oparta na ontologii fundacjonistycznej DOLCE. Układ ten powoduje, że opis ontologiczny jest bardzo nierównomierny, tzn. istnieją obszary dobrze opisane, tworzące regularną taksonomię, oraz obszary słabo opisane, ze strukturą wewnętrzną daleko odbiegającą od taksonomicznej. Wymienione tu zaburzenia taksonomii z jednej strony powodują poważne trudności z wykorzystaniem do wnioskowania algorytmu kartograficznego, z drugiej strony stwarzają bardzo dobre warunki do dokonania podziału ontologii PIPSDrugs na konteksty zawierające w sobie poszczególne grupy tematyczne.

Zaproponowany podział musiał spełnić wiele wymagań, jednak jednym z najważniejszych było obniżenie złożoności ontologii w sposób, który byłby istotny dla zastosowanego systemu KASEA. Aby ocenić wpływ podziału na tę złożoność, do jej wyrażenia zastosowana została miara zaproponowana w [GWW2008a]. Miara ta jest oparta na metodzie kartograficznej. Metoda ta tworzy mapy konceptów, które mogą być traktowane jako przedstawiciele zbioru dopuszczalnych modeli. Na podstawie tych map da się określić pewne właściwości

przetwarzanych ontologii. Innymi słowy, za pomocą pojęć mapy, dekompozycji i rozmieszczenia można zdefiniować pewien zestaw miar ontologicznych.



Rys. 7.2 Uproszczony schemat ontologii PIPSDrugs.

Zagadnienie kartograficznych miar ontologicznych jest szerzej dyskutowane w [GWW2008a], tutaj jednak warto wspomnieć, że w ogólności za pomocą pojęć kartograficznych można wyrazić dowolną miarę strukturalną (opisującą budowę ontologii), w dodatku tak zdefiniowane miary są semantyczne, czyli nie zależą od sposobu syntaktycznego wyrażenia zależności między terminami.

Z punktu widzenia niniejszej rozprawy szczególnie interesujący jest zestaw stosunkowo prostych miar, które są silnie skorelowane z wydajnością algorytmów wnioskujących z kartograficznej reprezentacji wiedzy. Najważniejszą miarą jest tutaj rozmiar mapy konceptów, czyli długość deskryptora binarnego. Miara ta wyraża pośrednio stopień złożoności ontologii (rozumianej jako możliwość przypisania osobnika do co najwyżej takiej liczby regionów mapy), zaś wraz z jej wzrostem efektywność wnioskowania spada, gdyż w jego trakcie należy uwzględniać więcej możliwości:

$$m(T) = \|M\| \quad (7.1)$$

W przypadku niepodzielonej ontologii PIPSDrugs liczba m wyniosła 23 787 766 176 ($2,38 \times 10^{10}$). Tak więc podstawowym zadaniem modularyzacji było zmniejszenie tej liczby.

7.2.2 Kontekstualizacja ontologii PIPSDrugs

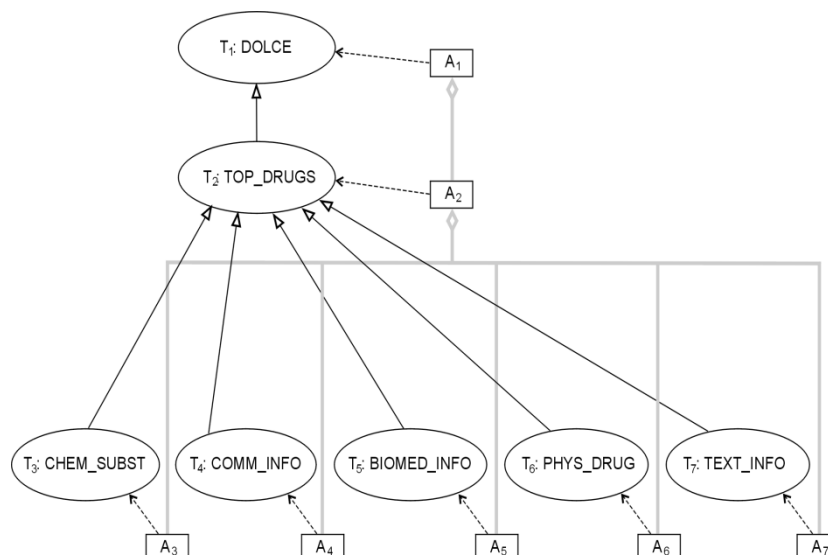
Pierwszym zadaniem związanym z podziałem ontologii PIPSDrugs było sformułowanie kryteriów, według których będzie można ocenić uzyskane efekty. Pierwsze z kryteriów zostało już opisane w poprzednim punkcie. Poddaje ono ocenie stopień zmniejszenia złożoności ontologii wyrażającej się miarą wielkości mapy. Kolejne kryteria skupiają się na jakości semantycznej dokonanego podziału oraz na różnego rodzaju problemach technicznych, których ten podział może być źródłem. Niewłaściwie przeprowadzona modularyzacja mogłaby bowiem spowodować utratę informacji lub utrudnić zadawanie zapytań. Dodatkowo, metoda SIM nie jest powszechnie przyjętym standardem, więc istotne było, aby zachować możliwość korzystania z ontologii w standardowy (niezmodularyzowany) sposób. Oprócz wymagań związanych z ewentualnością pogorszenia jakości bazy wiedzy lub utratą komfortu pracy, własne wymagania narzucił zastosowany system wnioskujący, którym był system KASEA. Głównym wymaganiem wynikającym z zastosowania tego systemu była konieczność translacji ontologii z języka OWL, w którym została wyrażona, do języka DIGUT.

Mając wszystkie powyższe uwarunkowania na uwadze, wprowadzone zostały następujące wymagania:

- Wprowadzony podział powinien dawać znaczny zysk w zakresie stopnia złożoności ontologii wyrażonego rozmiarem mapy konceptów.
- Konteksty powinny zawierać informacje, które nie są powiązane (np. poprzez subsumcję lub rolę) z informacjami z równoległych kontekstów. Rozdzielenie takich informacji wiązałoby się z utratą możliwości wyciągnięcia pewnych wniosków.
- Podział nie powinien wpływać na proces zadawania zapytań, tzn. żadne z zapytań, które byłoby zadane do ontologii niepodzielonej, nie powinno być rozbite na więcej zapytań tylko z powodu podziału na konteksty.
- Podział na konteksty ma być zrealizowany w ten sposób, aby jeden plik nie zawierał informacji z więcej niż jednego kontekstu.
- Należy utworzyć plik, który za pomocą polecenia `owl:import` połączy nowe pliki i umożliwi przeglądanie ontologii w całości.

- Powinna istnieć możliwość łatwego przetłumaczenia nowych plików na język DIGUT.

W wyniku przeprowadzonego podziału terminologia ontologii PIPSDrugs została podzielona na siedem typów kontekstów (patrz rys. 7.3). Do każdego z nich zostało podłączone jedno wystąpienie.



Rys. 7.3 Podział ontologii PIPSDrugs.

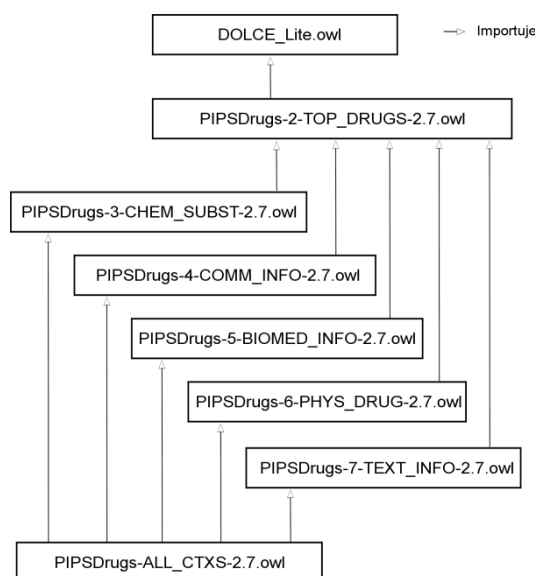
Ich nazwy i przeznaczenie zawarto w tabeli 7.1. Pięć typów *kontekstów tematycznych* jest związanych z odrębnymi grupami tematycznymi i zawiera definicje pojęć specyficznych dla tych grup. Stanowią one liście hierarchii. Typ kontekstu TOP_DRUGS definiuje koncepty i role, które są wspólne dla wszystkich kontekstów tematycznych. Typ DOLCE zawiera ontologię fundacjonistyczną, na której jest oparta ontologia PIPSDrugs.

Tabela 7.1 Podział ontologii PIPSDrugs na konteksty

Nazwa kontekstu	Przeznaczenie	Przykładowe koncepty
DOLCE	ontologia podstawowa	endurant perdurant quality
TOP_DRUGS	uszczegóławia pojęcia z ontologii podstawowej, zawiera pojęcia wspólne dla kontekstów tematycznych	biomedical-quality Chemical-Substance Drug-Prototype
CHEM_SUBST	opisuje skład chemiczny leków oraz możliwe interakcje między nimi; pozwala stwierdzić, czy dane dwa leki nie kolidują ze sobą	Active-Principle-Dosage Pharmaceutical-Form Drug-Interaction-Type
COMM_INFO	zawiera informację o handlowych właściwościach leków, głównie informacje o producencie i jego uprawnieniach	Company Marketing-Authorisation Patent
BIOMED_INFO	opisuje biomedyczne właściwości leków, informuje o dawkowaniu, o skutkach terapeutycznych i niepożądanych efektach	Shelf-Life Administration-Way Therapeutical-Effects
PHYS_DRUG	opisuje konkretne produkty oraz ich status, mówiący np. o tym, czy dana seria nie jest wycofana z rynku	Container EPC-Code Status
TEXT_INFO	opisuje informację na temat leków, zawartość ulotek, informatorów	Drug-Information-Type Information-Leaflet Therapeutic_indications

Każdy z typów i wystąpień kontekstów, opisany językiem OWL, został umieszczony w odrębnym pliku. Rysunek 7.4 przedstawia sposób łączenia plików zawierających terminologię za pomocą polecenia `owl:import`. Plik `PIPSDrugs-ALL_CTXS-2.7.owl` zawiera tylko polecenia importu, gdyż jego zadaniem jest umożliwienie przeglądania ontologii w całości za pomocą narzędzi takich jak *Protégé* (<http://protege.stanford.edu/>).

W razie konieczności, zmiany były wprowadzane do powyższych plików, a następnie pliki te były tłumaczone za pomocą translatora na język DIGUT. Wynikowe pliki translacji były ręcznie łączone i uzupełniane o rozkazy tworzące konteksty i ich wystąpienia.



Rys. 7.4 Rozmieszczenie kontekstów w plikach.

Zastosowany podział jest stosunkowo prosty. Nie oddaje on w pełni możliwości metody SIM. Główną tego przyczyną była konieczność oparcia się na edytorze *Protégé*, wynikająca z braku własnych narzędzi o podobnym przeznaczeniu uwzględniających kontekstową konstrukcję ontologii. Edytor ten wykazywał niekorzystne właściwości w sytuacji, gdy liczba importowanych plików OWL przekroczyła 5. Wprowadzenie kolejnych typów lub wystąpień kontekstów powodowało zawieszanie się systemu i uniemożliwiało dalszą pracę nad scenariuszem *Smart Cabinet*. W tej sytuacji podjęto decyzję, aby dokonany podział był jak najprostszy, lecz wystarczający, by osiągnąć znaczące zmniejszenie wielkości mapy konceptów.

7.2.3 Uzyskane wyniki

Głównym celem dokonanego podziału było obniżenie złożoności ontologii. Cel ten został z powodzeniem osiągnięty. Tabela 7.2 przedstawia wartości miary $m(T)$ dla poszczególnych kontekstów tematycznych. Widać, że liczba obszarów atomowych konceptu uniwersalnego radykalnie spadła. W przypadku najbardziej złożonego typu kontekstu wielkość mapy konceptów osiągnęła $m(T) = 2,3 \times 10^5$, czyli została zmniejszona o 5 rzędów wielkości.

Tabela 7.2 Złożoność kontekstów tematycznych wyrażona miarą $m(T)$

Nazwa kontekstu	$m(T)$
CHEM_SUBST	232887
COMM_INFO	35181

BIOMED_INFO	6847
PHYS_DRUG	6365
TEXT_INFO	14039

W trakcie dalszych prac okazało się, że wszystkie pozostałe z wymienionych w punkcie 7.2.2 wymagań zostały spełnione:

- Udało się rozdzielić koncepty i role w ten sposób, że nie utracono żadnej informacji zawartej w bazie wiedzy.
- Do wstępnego przetestowania wymagania dotyczącego wpływu podziału na proces zadawania zapytań, wykorzystano zestaw 22 zapytań kontrolnych. Mimo że przed zamknięciem scenariusza *Smart Cabinet* ulegały one wielokrotnym zmianom, ani razu nie zaistniała konieczność rozbijania zapytań między różne konteksty i składania cząstkowych wyników w jedną odpowiedź syntetyczną.
- Każdy plik zawierał dokładnie jeden kontekst.
- Plikami łączącymi wszystkie konteksty w całość był PIPSDrugs-ALL_CTXS-2.7.owl. Przeglądany w *Protégé* pozwalał obejrzeć ontologię w całości, choć wydajność tego narzędzia wyraźnie spadała.
- Translacja plików na język DIGUT nie nastąpiła z powodu zasadniczych trudności.

Zapytania kontrolne były formułowane przez zespół HSR, czyli twórców ontologii PIPSDrugs w wersji niezmodularyzowanej. W trakcie prac zestaw tych pytań był zmieniany, gdyż pojawiały się nowe wymagania wynikające z opracowywanego scenariusza *Smart Cabinet*. Zapytania były formułowane w języku naturalnym, po czym tłumaczono je na jeden ze standardowych problemów wnioskowania. Oto kilka przykładowych zapytań i ich wersja wyrażona w logice opisowej:

Jaki jest aktywny składnik leku Nuroxol?

Zapytanie do wystąpienia kontekstu CHEM_SUBST. Znajdź wszystkie znane wystąpienia konceptu:

```
Active-Principle ⊓
∃(hasIngredient- ∘ hasActivePrinciple- ∘
  hasPrototype-).∃(hasCommercialName ∘ hasDescription).="Nuroxol"
```

Jakie leki zawierające paracetamol są sprzedawane w kapsułkach?

Zapytanie do wystąpienia kontekstu BIOMED_INFO. Znajdź wszystkie znane wystąpienia konceptu:

```
Drug-Prototype ⊓ ∃(hasActivePrinciple ∘
  hasIngredient).{paracetamol} ⊓ ∃hasPharmaceuticalForm.{capsule}
```

Jaki są zamienniki leku Zovirax?

Zapytanie do wystąpienia kontekstu CHEM_SUBST. Znajdź wszystkie znane wystąpienia konceptu:

```
Commercial-Name ⊓ ∃(hasCommercialName- ∘ hasPrototype).∃(hasPrototype-
  ∘ hasCommercialName ∘ hasDescription).="Zovirax"
```

Czy lek o danym kodzie jest wycofany ze sprzedaży?

Zapytanie do wystąpienia kontekstu PHYS_DRUG. Czy osobnik o atrybucie *Code* równym *x* jest wystąpieniem konceptu:

$$EPC-Code \sqcap \exists Code. =_x \sqcap \\ \exists (hasCode^- \circ Contains^- \circ hasStatus \circ StatusValue). = "Withdrawn"$$

Użyty w powyższych wyrażeniach określających nazwę konceptu symbol \circ oznacza łańcuch ról, używany w formalizmie logiki opisowej oznaczanym literą \mathcal{R} . Zdarzały się również zapytania w języku naturalnym, które wymagały zadania większej liczby zapytań do bazy wiedzy, jednak ani razu nie było to spowodowane podziałem ontologii na konteksty. Przyczyną była po prostu kompleksowość zapytań.

7.3 Podsumowanie

W niniejszym rozdziale przedstawiono zagadnienia związane z praktycznym wdrożeniem metody w projekcie PIPS. Utworzenie odpowiednich mechanizmów wnioskujących odbyło się poprzez adaptację istniejącego systemu KASEA. Adaptacja ta pozwoliła na przetworzenie pierwszej praktycznej ontologii znacznych rozmiarów utworzonej za pomocą metody SIM i wykorzystanie jej w ramach jednego ze scenariuszy projektu PIPS.

Analizując proces tworzenia i eksploatacji ontologii PIPSDrugs, należy stwierdzić, że metoda SIM spełniła pokładane w niej nadzieje. Utworzony zestaw kontekstów okazał się intuicyjny i zrozumiały zarówno z punktu widzenia projektanta, jak i użytkownika. Co więcej zachowano możliwość odpowiedzi na zestaw wszystkich zapytań kontrolnych, przy jednoczesnym znacznym zmniejszeniu złożoności ontologii mierzonej liczbą regionów na mapie konceptów, dzięki czemu możliwe okazało się przyspieszenie procesu wnioskowania.

Wykorzystanie metody SIM nie zamknęło drogi do przetwarzania elementów ontologii za pomocą tradycyjnych narzędzi. Wyrażenie poszczególnych kontekstów w języku OWL pozwoliło na częściowe wykorzystanie edytorów ontologii w procesie wytwórczym, a także na użycie narzędzi służących m.in. do wizualizacji i walidacji ostatecznej wersji ontologii. Jednakże stwierdzono, że brak wyspecjalizowanych narzędzi jest dużym ograniczeniem uniemożliwiającym pełne wykorzystanie możliwości oferowanych przez opisywaną metodę.

Warto także zwrócić uwagę na fakt, że dokonany podział nie spowodował utraty informacji zawartej w ontologii. Potwierdza to główne założenie, które leży u podstaw metody SIM, że wiedza ma zawsze charakter kontekstowy. Wydzielenie kontekstów mogłoby więc zostać dokonane na początku prac projektowych, co z całą pewnością poprawiłoby organizację pracy i przyspieszyło wykonanie zadań.

8 Kierunki rozwoju metody SIM

Metoda SIM znajduje się obecnie w początkowej fazie rozwoju. Jednakże już w chwili obecnej ujawniła ona wiele zalet, które zachęcają do jej dalszego rozwijania. Należy tu wyróżnić dwa kierunki prac. Jeden kierunek to rozwój proponowanego przez metodę modelu kontekstowej bazy danych. W jego ramach podejmowane będą następujące zagadnienia: włączenie parametrów kontekstowych do formalizmu metody, dodanie nowych typów relacji, które zwiększą zakres właściwości dziedziny możliwych do ujęcia w strukturze bazy, opracowanie algebry kontekstowej oraz języka zapytań pozwalającego na tworzenie nowych, nie zdefiniowanych w schemacie bazy, kontekstów w trakcie fazy eksploatacji. Drugi kierunek to oprogramowanie wspierające fazę projektowania i implementacji oraz eksperymentalne systemy użytkowe nowego typu, które wykorzystywałyby wszystkie możliwości kontekstowych baz wiedzy. W dalszej części podrödziału wymienione wyżej tematy prac zostaną omówione szerzej.

Włączenie parametrów kontekstowych do formalizmu metody

W dotychczasowych rozważaniach silnie podkreślana była rola parametrów kontekstowych w procesie tworzenia i eksploatacji kontekstowej bazy wiedzy. Niestety, parametry kontekstowe jak dotąd nie zostały włączone do formalizmu metody. Istnieją tego poważne przyczyny. Przede wszystkim narzucałoby to konieczność ograniczenia ekspresywności parametrów w taki sposób, aby dało się opisać ich wpływ na definicję interpretacji, aby nie została wprowadzona nierozstrzygalność lub znaczące zwiększenie złożoności wnioskowania. Takie ograniczenie ekspresywności miałoby z pewnością ujemny wpływ na proces projektowania, gdyż uniemożliwiłoby wyrażanie bardziej skomplikowanych opisów kontekstów. Dodatkowo komplikowałoby to samą strukturę bazy wiedzy, gdyż zbiór parametrów, ich wartości, opis wpływu, jaki miałyby na wnioskowanie oraz sposób ich powiązania z pozostałymi elementami struktury musiałyby być zapamiętane. Niewątpliwie jednak włączenie wpływu przynajmniej części parametrów kontekstowych na przebieg procedury wnioskowania w znacznym stopniu zwiększyłoby możliwości modelowania. W opracowaniu [BBG2000] opisanym w punkcie 4.1.1 opisane są różne sposoby uwzględniania informacji zewnętrznej (czyli umieszczonej w parametrach). Jest to między innymi wstawianie i usuwanie (ang. *push and pop*) oraz przesuwanie (ang. *shifting*). Przesuwanie polega na tym, że znajomość parametrów kontekstowych i ich wartości na etapie wnioskowania pozwala ustanowić bardziej precyzyjne zasady przepływu wniosków z kontekstu do kontekstu. Na przykład gdyby w kontekście z datą 2.10.2010 znana była informacja z dnia 1.10.2010 o treści „dzisiaj pada”, mogłaby być interpretowana jako „wczoraj padało”. Wstawianie i usuwanie polega na tym, że informacja zewnętrzna jest czasem wstawiana do wnętrza kontekstu, a czasem usuwana z wnętrza kontekstu z powrotem do parametru. Wstawienie powinno skutkować powstaniem dodatkowych asercji w obrębie zawartości kontekstu i likwidacją parametru, a usunięcie ponownym pojawieniem się parametru i likwidacją odpowiadających mu asercji. W przykładzie 6.2 został wstawiony do kontekstu parametr *klasa* opisujący nazwę jednostki organizacyjnej, jest też rozważane wstawienie parametru *semestr*. To wszystko miało miejsce w fazie projektowania. Gdyby obsługa parametrów kontekstowych została włączona do metody SIM, takie wstawienie mogłoby zostać wykonane w fazie eksploatacji. Zwiększyłoby to dostęp użytkownika do informacji oraz poprawiło jakość wnioskowania.

Nowe typy relacji

Istniejące obecnie w ramach modelu danych udostępnianego przez metodę SIM relacje dziedziczenia, agregacji i konkretyzacji są wystarczające, aby opisać podstawowe zależności między kontekstami. Jednakże nie posiadają one wystarczającej ekspresywności, aby wyrazić

niektóre zjawiska znane z ludzkiego sposobu myślenia. Jedną z rozważanych relacji to *relacja aproksymacji*. Pozwalałaby ona łączyć kilka osobników w jeden przy przejściu w górę relacji agregacji. Działałoby to w ten sposób, że grupa osobników opisana na poziomie wystąpienia agregowanego jako części jakiejś całości nie byłaby przenoszona do sygnatury wystąpienia agregującego. W zamian byłby tam tworzony jeden osobnik reprezentujący grupę, który przejmowałby wszystkie role, które łączą osobniki należące do tej grupy z osobnikami spoza grupy. Z semantycznego punktu widzenia relacja ta dobrze oddawałaby zależność część-całość oraz pozwalałaby na lepsze zróżnicowanie wystąpień ze względu na poziom szczegółowości. Z punktu widzenia systemu dowodzenia przyczyniłoby się to do ograniczenia obszaru wnioskowania, gdyż nieobecność danych osobników w sygnaturze wystąpień ograniczyłaby przepływ wniosków.

Inną relacją, której dodanie jest rozważane, to *relacja reifikacji*, polegająca na tworzeniu związków między osobnikami w danym kontekście a wystąpieniami i typami kontekstów. Dzięki niej możliwa byłaby „samowiedza” kontekstowej bazy wiedzy. Taka samowiedza pozwoliłaby na wnioskowanie na metapoziomie, na przykład na opisanie stanu kontekstów roboczych w ramach kontekstów systemowych i udostępnienie tej informacji użytkownikowi. Możliwe byłoby wtedy opisanie mechanizmów kontrolnych, które ograniczyłyby niewłaściwe zachowania użytkowników, lub też wprowadzenie opisu uprawnień poszczególnych osób do różnych fragmentów struktury bazy. Najważniejszym celem wprowadzenia tej relacji byłoby jednak rozpoczęcie zupełnie nowego kierunku badań, którego celem byłoby opracowanie zasad definiowania dynamiki w obrębie bazy. Wydzielona kategoria typów kontekstów zawierałaby opisy procedur, które byłyby uruchamiane na skutek pewnych zdarzeń.

Algebra kontekstowa i język zapytań

W punkcie 6.1.3 wypowiedzianych zostało kilka uwag na temat tych działań użytkownika, które są związane z zadawaniem zapytań. Określono tam wymagania w stosunku do przyszłego języka. Język ten powinien pozwalać na zadawanie zapytań, których kompleksowość pozwoliłaby zgromadzić dane do wykonania raportu, jaki jest znany z systemów relacyjnych. Przygotowanie danych do raportów to oczywiście tylko jedna z funkcji języka zapytań. Wynik takiego zapytania mógłby również być kontekstem o typie nieistniejącym w schemacie bazy, utworzonym na chwilę na podstawie wiedzy pobranej z dowolnie wybranych kontekstów istniejących. Warunkiem powstania takiego języka jest opracowanie właściwej algebry kontekstów. Bez opracowania takiej algebry i bez stworzenia języka, o jakim mowa, nie będzie możliwe stworzenie nowej klasy systemów, o których mowa we wstępie do niniejszej rozprawy.

W opracowaniu odpowiedniej algebry pomoże autorowi doświadczenie, które uzyskał, uczestnicząc w opracowaniu algebry s-modułów ([GWW2009a], [GWW2009b], [GWW2009c], [GWW2009d]) oraz języka KQL ([GP2WZ2010a], [GP2WZ2010b]). Problem utworzenia właściwej algebry dla baz wiedzy jest bardzo specyficzny. Na ogół podejmowano próby opracowania algebry zdefiniowanej na zbiorze zdań ([D2008], [MW2004]). Niestety, takie opracowania najczęściej prowadziły do efektów rozmiągających się z oczekiwaniami. Przyczyną jest to, że, mówiąc w skrócie, zarówno pytania, jak i odpowiedzi muszą operować treściami. Jak wiadomo, dana treść może być wyrażona na wiele sposobów, czyli za pomocą wielu różnie zbudowanych zdań. Analiza tego problemu doprowadziła do wniosku, że poprawna algebra dla baz wiedzy nie powinna operować na zdaniach.

Algebra kontekstów nie operowałaby na zdaniach, lecz na klasach kontekstowych, które można byłoby zdefiniować jako pary $C = (T, \{A \mid A \in E[A_T] \wedge E \in \mathbf{E}_H\})$, gdzie H jest danym schematem, a $T \in H[\mathbf{T}]$. Tak zdefiniowana klasa kontekstowa składałaby się z typu

kontekstu i wszystkich jego wystąpień w ramach jednej instancji danego schematu. Można odnaleźć pewną analogię klasy kontekstowej do relacji, która składa się z nagłówka i krotek. Ponieważ kontekst został w podrozdziale 5.1 zdefiniowany jako para składająca się z typu i wystąpienia, można uważać klasę kontekstową za zbiór kontekstów. Dana operacja algebry kontekstów będzie przekształcała daną klasę kontekstową (operator unarny) lub parę klas kontekstowych (operator binarny) w nową klasę kontekstową. Aby opisać typ nowej klasy kontekstowej (projekcję) oraz ograniczyć zawartość zbioru wystąpień (selekcję), trzeba będzie sformułować odpowiednie ograniczenia w logice opisowej. Takie właśnie rozwiązanie zostało zastosowane z powodzeniem w algebrze konglomeratów.

Oprogramowanie wspierające fazę projektowania i implementację

Obecnie nie istnieje żadne oprogramowanie wspierające projektowanie kontekstowych baz wiedzy. Opisaną w 7.2.2 konwersję ontologii PIPSDrugs na wersję kontekstową autor przeprowadzał ręcznie z wykorzystaniem zwykłego edytora tekstu. Jest to bardzo uciążliwe. Pierwszym zadaniem musi więc być opracowanie narzędzia podobnego do *Protégé*, najbardziej obecnie znanego edytora ontologii płaskich (<http://protege.stanford.edu/>). Oprogramowanie tego rodzaju można byłoby wyposażać w wiele urządzeń wspomagających projektanta w kwestiach merytorycznych. Na przykład zaimplementowanie rozwiązania zaproponowanego we wspomnianych już pracach [GLW2006], [CHKS2007] i [GHKS2008] pomogłoby w utrzymaniu zasad zachowania modelu z definicji 5.10, co zabezpieczyłoby tworzoną ontologię przed niekoherencjami. Z kolei wykorzystanie zaproponowanego w [HS2005] pojęcia minimalnej niespójnej subontologii (*minimal inconsistent subontology* — *MIS*) pozwoliłoby wykryć przyczynę ewentualnych niespójności.

Jednakże stworzenie nowej klasy systemów, o których mowa we wstępie do niniejszej rozprawy, wymaga przede wszystkim opracowania wielodostępowego serwera wiedzy, którego zadania byłyby podobne do współczesnych systemów typu DBMS. Opracowanie takiego serwera wiąże się z koniecznością rozwiązania wielu problemów, na przykład problemu transakcji, które musiałyby mieć podobne właściwości, jak transakcje w systemie relacyjnym.

Oprogramowanie wspierające implementację systemów proponowanej klasy to przede wszystkim odpowiednie sterowniki do serwerów bazy wiedzy, które zapewnią programiście interfejs pozwalający formułować zapytania i otrzymywać odpowiedzi. Opracowanie takiego oprogramowania wiąże się jednak z rozwiązaniem poważnego problemu przeniesienia wiedzy zawartej w wystąpieniach kontekstów do środowiska obiektowego. Należy pamiętać, że nastąpi tu zderzenie koncepcji świata otwartego z koncepcją świata zamkniętego. Oznacza to, że dany kontekst zawiera nie tylko przyporządkowane mu formuły logiczne, lecz także wszystkie wnioski, które wynikają z tych formuł i z powiązań tego kontekstu z innymi kontekstami. Tak więc po zadaniu zapytania, gdy użytkownik otrzyma w odpowiedzi klasę kontekstową, konieczne będzie odpowiednie opisanie zakresu i formy informacji, która powinna być z niej wydobyta. Opracowanie odpowiednich sposobów takiego opisu powinno być tematem poważnych badań naukowych.

9 Podsumowanie

Celem każdej rozprawy doktorskiej jest dowiedzenie tezy, która w przypadku niniejszej pracy brzmi:

Organizacja ontologii w formie hierarchii kontekstów powoduje uproszczenie ontologii i usprawnienie pracy systemów wnioskujących, a także ułatwia inżynierom ontologii konstruowanie złożonych baz wiedzy i bardziej adekwatne modelowanie w nich istotnych problemów występujących w inżynierii wiedzy, takich jak reprezentowanie różnych punktów widzenia, rozwiązywanie antynomii, aktualizowanie baz wiedzy przy zachowaniu monotoniczności i in.

Teza ta, wypowiedziana w sposób ogólny, została we wstępie uszczegółowiona i odniesiona do konkretnych problemów występujących w pracach dotyczących reprezentacji wiedzy i wnioskowania. Autor ze swojej strony dodał własne wymagania, które wypływają z pragnienia realizacji jego własnej wizji upowszechnienia metody ontologicznej i uczynienia z niej alternatywy dla obecnych metod realizacji systemów informatycznych, w szczególności metody relacyjnej i obiektowej. Idea narzucenia ontologii hierarchicznej struktury kontekstowej znalazła swój wyraz w opracowanej przez autora metodzie SIM. W rezultacie powstały dwie tezy pomocnicze, które swą treścią odpowiadają tezie podstawowej, a jednocześnie konkretyzują występujące tam zagadnienia:

Teza pomocnicza 1:

Metoda SIM pozwala rozwiązać problem konstrukcyjny, organizacyjny i wydajnościowy w podobny lub lepszy sposób, jak inne metody modularyzacji.

Teza pomocnicza 2:

Metoda SIM pozwala rozwiązać problem adekwatności kontekstów i problem projektowania kontekstowego.

Aby dowieść powyższych tez, właściwości metody SIM zostały przedstawione na tle ogólnych rozważań na temat konceptualizacji (patrz rozdziały 2 i 3) oraz wielu różnych prac dotyczących kontekstowego sposobu reprezentowania wiedzy (rozdział 4). Najważniejszą rolę w budowaniu tego tła odgrywają prace N. Guarino (opisane w punkcie 2.1.4), metoda walidacji ontologii OntoClean (podrozdział 3.1) i rozważania o wnioskowaniu w środowisku kontekstowym (punkt 4.1.1).

Hierarchiczna kontekstualizacja ontologii została zaprezentowana obok innych metod modularyzacji ontologii, które kierowały się podobnymi celami i proponowały podobne rozwiązania (DDL i \mathcal{E} -Connections, patrz punkty 4.2.1 i 4.2.2). Skonfrontowanie metody SIM z tymi metodami pozwala uwypuklić cechy, które stanowią podstawę dla dowiedzenia wypowiedzianych powyżej tez. Tabela 9.1 opisuje te cechy:

Tabela 9.1 Cechy metody SIM na tle innych metod modularyzacji

	Metoda SIM	Inne metody modularyzacji
1)	Powiązania między modułami mają charakter hierarchiczny.	Moduły są wiązane każdy z każdym w nieuporządkowaną sieć.
2)	Wyróżnione są dwa typy modułów: terminologiczne, nazywane typami kontekstów, i zawierające opis świata, nazywane wystąpieniami kontekstów.	Moduły są nierozróżnialne, są pełnymi ontologiami zawierającymi Tbox i Abox.

3)	Jeden typ kontekstu może być powiązany z wieloma wystąpieniami za pomocą relacji konkretyzacji.	Brak możliwości wiązania wielu Aboxów z jednym Tboxem.
4)	Moduły różnych typów są porządkowane różnymi typami relacji: typy kontekstów są porządkowane za pomocą relacji dziedziczenia, a wystąpienia za pomocą relacji agregacji.	Istnieje tylko jeden typ relacji.
5)	Obie relacje posiadają ściśle zdefiniowaną semantykę, niewymagającą dodatkowych opisów, która determinuje zależności między zawartościami łączonych modułów.	Relacje między modułami nic nie mówią o semantyce. Konieczny jest dodatkowy opis, relacje pomostowe lub relacje łączące, który specyfikuje zależności między zawartościami modułów.
6)	Struktura bazy wiedzy semantycznie odzwierciedla właściwości opisywanego problemu odpowiednio do analizy wykonanej za pomocą parametrów kontekstowych.	Podział na moduły nie odzwierciedla właściwości opisywanego problemu.
7)	Właściwości metody pozwalają na oddzielenie schematu bazy od jej instancji. Dzięki temu można oddzielić etap projektowania bazy od etapu eksploatacji.	Brak możliwości podziału na schemat i instancję bazy. Nie istnieje możliwość wyodrębnienia etapu projektowania i etapu eksploatacji bazy wiedzy.
8)	Istnieje możliwość odzwierciedlenia niektórych metawłaściwości konceptów, na przykład immanentności lokalnej, w strukturze bazy.	Metawłaściwości konceptów mają jedynie zasięg lokalny i nie wpływają na przepływy wniosków między modułami.

Powyższe cechy sprawiają, że SIM dostarcza inżynierom wiedzy narzędzi pozwalających prostymi środkami wyrazić związki między modułami. Zamiast tworzyć precyzyjne, ale bardzo pracochłonne i bardzo trudne do modyfikowania zbiory reguł pomostowych lub reguł łączących, wystarczy zastosować proste w realizacji relacje dziedziczenia i agregacji. Ta prostota wcale nie zmniejsza możliwości wyrażania złożonych zależności między modułami. W punktach 6.2.1 i 6.2.2 przedstawiono, w jaki sposób za pomocą różnych układów kontekstów można wyrazić skomplikowane relacje między zawartymi w nich fragmentami wiedzy. Oznacza to, że metoda SIM rozwiązuje problem konstrukcyjny i organizacyjny. Robi to nie gorzej od innych metod modularyzacji. Jednak najważniejszą zaletą metody SIM jest to, że, w odróżnieniu od innych metod, przypisuje modułom semantykę pozwalającą interpretować je na poziomie dziedziny. Dzięki temu inżynier wiedzy zyskuje dodatkowe środki wyrazu, która wzbogacają opis dziedziny wyrażany przez formuły logiczne zawarte w bazie wiedzy. To sprawia, że uzasadnione jest stwierdzenie, że metoda SIM rozwiązuje problem konstrukcyjny i organizacyjny lepiej niż inne metody. Ontologia uporządkowana hierarchicznie jest dużo bardziej czytelna i łatwiejsza do ogarnięcia niż nieuporządkowana sieć modułów łączonych każdy z każdym. Z kształtu połączeń hierarchicznych wynika od razu czytelna semantyka, która w przypadku innych metod jest dostępna dopiero po zaznajomieniu się z treścią reguł pomostowych lub łączących.

Silnym argumentem przemawiającym za prawdziwością powyższych stwierdzeń jest porównanie metody SIM z innymi w świetle zasad kontekstowego wnioskowania przedstawionych w [BBG2000] a opisanych w punkcie 4.1.1:

Tabela 9.2 Porównanie metody SIM z innymi metodami w świetle zasad kontekstowego wnioskowania

Postulat wnioskowania kontekstowego z [BBG2000]	Metoda SIM	Inne metody modularyzacji
Kontekst jako pudełko opisane za pomocą parametrów kontekstowych	Struktura bazy wiedzy jest adekwatna do struktury kontekstowego opisu parametrycznego	Parametry kontekstowe nie są używane, struktura bazy wiedzy jest nieuporządkowana

Zakres kontekstu	W dół hierarchii dziedziczenia zakres jest coraz mniejszy, maleje dziedzina, maleje liczba opisywanych osobników	To, w jakiej relacji względem siebie znajdują się dziedziny modułów wynika jedynie z treści reguł pomostowych lub łączących.
Poziom szczegółowości	W dół hierarchii dziedziczenia szczegółowość jest coraz większa, co wyraża się coraz bardziej precyzyjną terminologią	Wynika jedynie z treści reguł pomostowych lub łączących.
Perspektywa	Wynika z kontekstowego opisu parametrycznego, który ma swoje odzwierciedlenie w kształcie relacji agregacji. Wystąpienia agregowane przedstawiają różne perspektywy ujęcia różnych fragmentów wystąpienia agregowanego	j.w.
Reguła lokalności wnioskowania kontekstowego	Spełniona	Spełniona
Reguła kompatybilności wnioskowania kontekstowego	Spełniona	Spełniona

Z powyższego zestawienia widać wyraźnie, że metoda SIM w sposób dużo bardziej adekwatny wychodzi naprzeciw postulowanym w [BBD2000] cechom wnioskowania kontekstowego. Jest to zasługą semantyki przypisanej poszczególnym elementom struktury bazy wiedzy. Inne metody całą semantykę ukrywają w regułach pomostowych lub łączących. Zważywszy, że opis ukryty w tych regułach jest trudno czytelny i trudno dostępny, problem konstrukcyjny jest łagodzony w dużo mniejszym stopniu.

W punkcie 6.1.4 struktura kontekstów, parametryczny opis kontekstowy oraz niektóre zasady metody walidacji OntoClean zostały połączone w jedną całość. Powiązanie w jakąś całość tak wydawałoby się odległych dziedzin tematycznych, jak pudełkowa metafora kontekstu i metoda OntoClean, której celem jest znormalizowanie procesu konceptualizacji, świadczy o tym, że metoda SIM ma również walor unifikujący. Pozwala to przerzucić pomost między kontekstowym i niekontekstowym sposobem widzenia opisywanej dziedziny i umożliwia wyrażenie ich wzajemnych odniesień. Z przeprowadzonych rozważań wynika, że kontekstowość ontologii polega przede wszystkim na możliwości ujęcia w ramach jednego opisu całej „wiązki” światów możliwych. Znajduje to wyraz w tym, że ontologia kontekstowa zachowuje monotoniczność w sytuacji, gdy ontologia płaska musi ją utracić. To wyznacza kierunki działań, które zostały we wstępie do tej rozprawy nazwane projektowaniem kontekstowym. Projektowanie kontekstowe to takie projektowanie, w trakcie którego projektant określa dopuszczalną wiązkę światów możliwych i wyraża ją przez odpowiednio dobraną strukturę kontekstów.

W ten sposób metoda SIM rozwiązuje problem adekwatności kontekstów i problem projektowania kontekstowego.

Dowiedzenie przydatności opisywanej metody do rozwiązania problemu wydajnościowego opiera się na dwóch spostrzeżeniach. Spostrzeżenie pierwsze dotyczy wnioskowania terminologicznego. Bez względu na zastosowany algorytm, wnioskowanie terminologiczne z danego kontekstu musi być mniej kosztowne niż wnioskowanie z całej ontologii, gdyż obejmuje jedynie aksjomaty obecne w terminologii tego kontekstu. Spostrzeżenie drugie dotyczy wnioskowania ze zbioru faktów. W tym przypadku swoją przewagę nad innymi, szczególnie dla dużych Aboxów, wykazał algorytm kartograficzny, co zostało dowiedzione w pracy [W2007]. Jego wersja, nazwana algorytmem strukturalnym,

uwzględniająca zasady metody SIM, została przedstawiona w punkcie 5.3.3. Wersja ta została zaimplementowana w systemie KASEA opisanym w podrozdziale 7.1 oraz zastosowana praktycznie w ramach projektu PIPS do wnioskowania z ontologii PIPSDrugs (patrz podrozdział 7.2). Uzyskane wyniki dowiodły, że zastosowanie metody SIM obniżyło złożoność wnioskowania, wyrażoną wartością miary $m(T)$, o 5 rzędów wielkości.

Oba te spostrzeżenia pozwalają stwierdzić, że metoda SIM rozwiązuje problem wydajnościowy, przy czym podczas wnioskowania terminologicznego daje rezultaty podobne do innych przedstawionych metod, natomiast podczas wnioskowania ze zbioru faktów silnie wspiera metodę kartograficzną posiadającą nad innymi metodami dużą przewagę.

Przechodząc do mankamentów zaproponowanego rozwiązania, należy przede wszystkim stwierdzić, że parametry kontekstowe nie są jeszcze częścią formalizmu metody SIM, a jedynie są zalecane jako pewna praktyka projektowa. To powoduje, że nie jest możliwe uwzględnianie informacji zewnętrznej (zawartej w parametrach), takie jak opisane w punkcie 4.1.1 wstawianie i usuwanie oraz przesuwanie, podczas wnioskowania. Jest to dość poważne ograniczenie relacji agregacji: wystąpienie kontekstu, syntetyzując wiedzę z wystąpień agregowanych, „nie zna” przyczyny agregacji. Nie jest więc możliwa synteza uzależniona od znaczenia parametrów — działa jednorodny mechanizm, który traktuje wszystkie rodzaje agregowanych wystąpień tak samo.

Kolejnym minusem jest brak mechanizmów tworzenia zabezpieczeń przed niewłaściwym zachowaniem osobę eksploatującą bazę wiedzy. Nie istnieje pojęcie będące odpowiednikiem kluczy kandydujących lub związków encji, które utrudniłyby na przykład dublowanie wystąpień kontekstów lub uzależnianie informacji umieszczonej w równoległych ścieżkach dziedziczenia.

Wymienione wady nie są jednak wadami substancjalnymi, lecz raczej akcydentalnymi. Wynikają z tego, że metoda SIM znajduje się dopiero w początkowej fazie rozwoju. Dadzą się one z łatwością usunąć w kolejnych jej wersjach. Część z nich może zostać usunięta w ramach prac związanych z realizacją planów dalszego rozwoju metody opisanych w rozdziale 8. W związku z tym nie powinny one negatywnie wpłynąć na ocenę przedstawionego rozwiązania. W opinii autora należy więc uznać, że przydatność metody SIM do rozwiązania problemów zawartych w obu tezach pomocniczych została udowodniona, a zatem główna teza rozprawy może być również uznana za dowiedzioną.

W opinii autora, najistotniejsze osiągnięcia niniejszej rozprawy są następujące:

- **Zaproponowanie nowej metody modularyzacji ontologii SIM** polegającej na tworzeniu hierarchicznych struktur kontekstów połączonych relacjami dziedziczenia oraz agregacji.
- **Dokonanie analizy zagadnień związanych z wnioskowaniem**, w ramach której opisano dwa przykładowe algorytmy wnioskowania (integracyjny i strukturalny).
- **Przeprowadzenie analizy właściwości metody SIM**, z wykazaniem, że harmonizuje ona parametryczne ujęcie semantyki kontekstów oraz zasady wnioskowania kontekstowego przedstawione w [BBG2000] z normami projektowymi narzucanymi przez metodę OntoClean.
- **Weryfikacja tej metody praktycznie** w ramach projektu PIPS finansowanego w ramach 6. Programu Ramowego Unii Europejskiej.
- **Skonfrontowanie metody SIM z innymi metodami modularyzacji ontologii**: DDL i \mathcal{E} -Connections, ze wskazaniem lepszego przystosowania tej pierwszej do

rozwiązania problemów projektowych i eksploatacyjnych w tym specyficznych dla kontekstowego ujmowania wiedzy.

- **Wytyczenie kierunków rozwoju**, które pozwoliłyby wykorzystać potencjał tkwiący w metodzie do szerszego jej zastosowania w systemach informatycznych powszechnego użytku.

Podsumowując, metoda SIM, już w obecnym stadium, stanowi nowatorską propozycję pozwalającą na systematyczne ujęcie wielu problemów związanych z wykorzystaniem ontologii w praktyce inżynierskiej. Można mieć nadzieję, że przyszły jej rozwój zaowocuje opracowaniem procesu wytwórczego, który, wspierany przez odpowiednią infrastrukturę narzędziową, znacznie rozszerzy możliwości wdrożenia ontologii i związanych z nimi metod sztucznej inteligencji do szerokiej gamy systemów informatycznych.

Bibliografia

- [BMNP2003] Baader F. A., McGuinness D. L., Nardi D., Patel-Schneider P. F. (red.): The Description Logics Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.
- [BC2009] Bąk J., Cybulka J.: Metoda przyrostowego tworzenia bazy wiedzy na podstawie bazy danych, ontologii i wiedzy eksperta. W: *Studia Informatica*, vol. 30, no 2A (83), 2009, s. 169-185.
- [B2005] Ben-Ari M.: *Logika matematyczna w informatyce*. WNT, Warszawa, 2005.
- [BBG2000] Benerecetti, M., Bouquet, P., Ghidini, C.: Contextual reasoning distilled. W: *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 12, Lipiec 2000, s. 279-305.
- [BBG2001] Benerecetti, M., Bouquet, P., Ghidini, C.: On the dimensions of context dependence: Partiality, approximation, and perspective. W: *Proceedings of CONTEXT 2001*.
- [BHL2001] Berners-Lee T., Hendler J., Lassila O.: The Semantic Web. *Scientific American*, nr 5, 2001, s. 34-43.
- [BS2003] Borgida A., Serafini L.: Distributed Description Logics: Assimilating Information from Peer Sources. W: *Journal on Data Semantics*, 2003, s. 153-184.
- [BS2002] Borgida A., Serafini L.: Distributed Description Logics: Directed Domain Correspondence in Federated Information Sources. W: *On the Move to Meaningful Internet Systems 2002*, LNCS, Vol. 2519, Springer Verlag, 2002, s. 36-53.
- [BL2004] Brachman R. J., Levesque H. J.: *Knowledge representation and reasoning*, Morgan Kaufman Publishers, San Francisco, 2004.
- [BBM1994] Buvač, S., Buvač, V. i Mason, I.: The semantics of propositional contexts. W: *Methodologies for Intelligent Systems*, Springer Verlag, Berlin, Heidelberg, 1994, s. 468-477.
- [C1979] Chisholm R. M.: Identity through Possible Worlds: Some Questions. W: Loux M. J. (red.): *The possible and the actual: readings in the metaphysics of modality*. Cornell University Press, 1979, s. 80-87.
- [C1972] Codd E. F.: Further Normalization of the Data Base Relational Model. W: *Data Base Systems*, Courant Computer Science Symposia Series 6, Englewood Cliffs, Prentice-Hall, New York, 1972.
- [CHKS2007] Cuenca Grau B., Horrocks I., Kazakov Y., Sattler U.: A logical framework for modularity of ontologies. W: *Proceedings of IJCAI-2007*, 2007, s. 298-304.
- [CGY2007] Cullot, N., Ghawi, R., and Yetongnon, K. : DB2OWL: A Tool for Automatic Database-to-Ontology Mapping. In *Proceedings of 15th Italian Symposium on Advanced Database Systems (SEBD 2007)*, Torre Canne, Włochy, 2007, s. 491-494.
- [CP2007] Cybulka J., Meissner A., Pankowski T.: Zastosowanie ontologii w procesie odwzorowywania schematów i wymiany danych XML. W: *Kozielski S.*,

- Małysiak B., Kasprowski P., Mrozek D. (red.): Bazy danych. Nowe technologie. Wydawnictwa Komunikacji i Łączności, Warszawa 2007, s. 69-80.
- [DSM2006] d'Aquin M., Sabou M., Motta E.: Modularization: a Key for the Dynamic Selection of Relevant Knowledge Components. W: Proceedings of ISWC 2006 Workshop on Modular Ontologies (WoMO 2006).
- [D2008] d'Aquin M. (red.): NeOn Formalisms for Modularization: Syntax, Semantics, Algebra. Neon Project, Deliverable D1.1.3, 2008.
- [D2000] Date C. J.: Wprowadzenie do systemów baz danych. Wydanie II, WNT, Warszawa, 2000.
- [DGMS2006] Domínguez D., Grasso F., Miller T., Serafin R.: PIPS: An Integrated Environment for Health Care Delivery and Healthy Lifestyle Support. W: Moreno A., Cortes U., Annicchiarico R., Nealon J. (red.): Working notes of the 4th Workshop on Agents Applied in Health Care, ECAI 2006. Riva del Garda, 2006.
- [F1982] Forgy C. L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. W: Artificial Intelligence, Vol. 19, nr 1, 1982, s. 17-37.
- [GSG1996] Gangemi A., Steve G., Giacomelli F.: ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration. W: Proceedings of ECAI-96 Workshop on Ontological Engineering. Budapest, 1996.
- [G2006] Garson J.W.: Modal Logic for Philosophers. Cambridge University Press, 2006.
- [GN1987] Genesereth M. R., Nilsson N. J.: Logical Foundation of Artificial Intelligence. Morgan Kaufmann, Los Altos, California, 1987.
- [GG2001] Ghidini C., Giunchiglia F.: Local model semantics, or contextual reasoning = locality + compatibility. Artificial Intelligence, 127(2), 2001, s. 221-259.
- [GS1998] Ghidini C., Serafini L.: Distributed First Order Logics. W: Gabbay D., de Rijke M. (red.): Frontiers of Combining Systems 2, Studies in Logic and Computation, Research Studies Press, 1998, s. 121-140.
- [GLW2006] Ghilardi S., Lutz C., Wolter F.: Did I damage my ontology? A case for conservative extensions in description logics. W: Proceedings of KR-06, 2006.
- [GS1994] Giunchiglia F., Serafini L.: Multilanguage hierarchical logics, or: how we can do without modal logics. W: Artificial Intelligence, 65(1), 1994, s. 29-70.
- [GP2WZ2010a] Goczyła K., Piotrowski P., **Waloszek A.**, Waloszek W.: Język KQL jako realizacja idei języka SQL dla bazy wiedzy. W: Studia informatica, Vol. 31, Nr 2A (89), 2010, s. 126-137.
- [GP2WZ2010b] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T.: KQL as Application of SQL Rationale for Knowledge Bases. W: Zeszyty naukowe wydziału elektroniki, telekomunikacji i informatyki Politechniki Gdańskiej, Vol. 18, 2010, s. 69-74.

- [GP2W2Z2008] Goczyła K., Piotrowski P., **Waloszek A.**, Waloszek W., Zawadzka T., Zawadzki M.: Zarządzanie wiedzą ontologiczną w środowisku semantycznego Internetu. W: Huzar Z., Mazur Z. (red.): Inżynieria oprogramowania - od teorii do praktyki, Rozdz. 25., red., WKŁ, Warszawa 2008, s. 289-302.
- [GW2008a] Goczyła K., **Waloszek A.**: A Distributed Description of Facts in a Distributed Semantic Web Ontology. W: Proceedings 1st IEEE International Conference on Information technology IT 2008, 19-21 May 2008, Gdańsk, 2008, s. 39-42.
- [GW2008b] Goczyła K., **Waloszek A.**: Rozproszony opis świata w rozproszonych ontologiach Sieci Semantycznej. Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, Technologie Informacyjne, Tom 15, Politechnika Gdańska, Gdańsk, 2008, Red.: A. Stepnowski, M. Moszyński, str. 7-12.
- [GWW2006] Goczyła K., **Waloszek A.**, Waloszek W.: Concrete-domain Reasoning Techniques in Knowledge Cartography. W: Semantics Systems From Visions to Application, Oesterreiche Computer Gesellschaft, 2006, s. 127-138.
- [GWW2007a] Goczyła K., **Waloszek A.**, Waloszek W.: Contextualization of a DL knowledge base. W: Proceedings of the 20th International Workshop on Description DL'07, Brixen/Bressanone, Włochy, 2007, s. 291-299.
- [GWW2007b] Goczyła K., **Waloszek A.**, Waloszek W.: Hierarchiczny podział przestrzeni ontologii na konteksty. W: Kozielski S., Małysiak B., Kasprowski P., Mrozek D. (red.): Bazy danych. Nowe technologie - Architektura, metody formalne i zaawansowany analiza danych. Wydawnictwa Komunikacji i Łączności, Warszawa, 2007, s. 247-260.
- [GWW2007c] Goczyła K., **Waloszek A.**, Waloszek W.: Wnioskowanie z dziedzin konkretnych w podejściu kartograficznym. W: Kozielski S., Małysiak B., Kasprowski P., Mrozek D. (red.): Bazy danych. Nowe technologie - Architektura, metody formalne i zaawansowany analiza danych. Wydawnictwa Komunikacji i Łączności, Warszawa, 2007, s. 261-273.
- [GWW2008a] Goczyła K., **Waloszek A.**, Waloszek W.: Pomiar właściwości ontologii za pomocą metody kartograficznej. W: Kozielski S., Małysiak B., Kasprowski P., Mrozka D. (red.): Bazy danych. Rozwój metod i technologii, WKŁ, 2008, s. 295-308.
- [GWW2008b] Goczyła K., **Waloszek A.**, Waloszek W.: Techniki modularyzacji ontologii. W: Kozielski S., Małysiak B., Kasprowski P., Mrozka D. (red.): Bazy danych. Rozwój metod i technologii, WKŁ, 2008, s. 310-322.
- [GWW2009a] Goczyła K., **Waloszek A.**, Waloszek W.: A Semantic Algebra for Modularized Description Logics Knowledge Bases. W: Cuenca Grau B. (red.): The 22nd International Workshop on Description Logics (DL2009). Oxford, 2009.
- [GWW2009b] Goczyła K., **Waloszek A.**, Waloszek W.: Algebra konglomeratów jako narzędzie opisu problemów przetwarzania ontologii. W: Studia Informatica, Zeszyty naukowe Politechniki Śląskiej. Gliwice: Silesian University of Technology Press, Vol. 30, nr 2A/83 (2009), s. 141-156.

- [GWW2009c] Goczyła K., **Waloszek A.**, Waloszek W.: Algebra of ontology modules for semantic agents. W: Nguyen N. T., Kowalczyk R., Chen S. M. (red.): Computational collective intelligence: semantic web, social networks and multiagent systems. LNAI, Vol. 5796, Springer Verlag, 2009, s. 492-503.
- [GWW2009d] Goczyła K., **Waloszek A.**, Waloszek W.: S-Modules – An Approach to Capture Semantics for Modularized DL Knowledge Bases. W: KEOD 2009, proceedings of the international conference on knowledge engineering and ontology development, Funchal-Madeira, Portugal, October 6-8, 2009, s. 117-122.
- [GWW2009e] Goczyła K., **Waloszek A.**, Waloszek W.: Ontologia PIPSDrugs: praktyczne zastosowanie metody SIM. W: . W: Studia informatica, Vol. 30, Nr 2A (83), 2009, s. 157-168.
- [GWWZ2010a] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T.: Ideologiczny i praktyczny model metaontologii. W: Studia informatica, Vol. 31, Nr 2A (89), 2010, s. 63-75.
- [GWWZ2010b] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T.: Ideological and Practical Model of Metaontology. W: Zeszyty naukowe wydziału elektroniki, telekomunikacji i informatyki Politechniki Gdańskiej, Vol. 18, 2010, s. 75-81.
- [GWWZ2010c] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T.: Internal Representation of S-Modular Knowledge Base in RKaSeA System. W: Zeszyty naukowe wydziału elektroniki, telekomunikacji i informatyki Politechniki Gdańskiej, Vol. 18, 2010, s. 63-68.
- [GWWZ2010d] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T.: Wewnętrzna reprezentacja konglomeratowej bazy wiedzy w systemie RKaSeA. W: Studia informatica, Vol. 31, Nr 2A (89), 2010, s. 105-120.
- [GWWZ2010e] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T.: Wnioskowanie z różnych źródeł osobników w systemie RKaSeA. W: Studia informatica, Vol. 31, Nr 2A (89), 2010, s. 89-103.
- [G2W2Z2006] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T., Zawadzki M.: Ontological Queries Supporting Decision Process in KaSeA System. W: Proceedings of 16th European-Japanese Conference on Information Modelling and Knowledge Bases, Ostrawa 2006, s. 16-28.
- [G2W2Z2007] Goczyła K., **Waloszek A.**, Waloszek W., Zawadzka T., Zawadzki M.: Ontological queries supporting decision process in KaSea system. W: Information Modelling and Knowledge Bases XVIII (Frontiers in Artificial Intelligence and Applications), IOS Press, 2007, s. 17-29.
- [G2004] Graham I.: Metody obiektowe w teorii i w praktyce. WNT Warszawa, 2004.
- [GHKS2008] Grau B. C., Horrocks I., Kazakov Y., Sattler U.: Modular Reuse of Ontologies: Theory and Practice. W: Journal of Artificial Intelligence Research (JAIR), Vol. 31, 2008, s. 273–318.
- [GPS2004] Grau B. C., Parsia B., Sirin E.: Tableau Algorithms for \mathcal{E} -Connections of Description Logics. Raport badawczy UMIACS, Maryland, 2004.

- [GHA2007] Grimm S., Hitzler P., Abecker A.: Knowledge Representation and Ontologies. W: Studer R., Grimm S., Abecker A. (red.): Semantic Web Services. Springer Verlag 2007, s. 51-105.
- [GHVD2003] Grosz B., Horrocks I., Volz R., Decker S.: Description Logic Programs: Combining Logic Programs with Description Logic. W: SSRN Electronic Journal, 2003.
- [G1995] Gruber T. R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies, 43(5/6), 1995, s. 907-928.
- [GWPZ2004] Gu T., Wang X. H., Pung H. K., Zhang D. Q.: An Ontology-based Context Model in Intelligent Environments. W: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, 2004, s. 270-275.
- [G1997] Guarino N.: Some Organizing Principles For a Unified Top-Level Ontology. W: Proceedings of the AAAI Spring Symposium On Ontological Engineering, Stanford, USA, 1997.
- [G1998] Guarino N.: Formal Ontology and Information Systems. W: Proceedings of the 1st International Conference on Formal Ontologies in Information Systems FOIS'98. Trento, Italy, IOS Press, 1998, s. 3-15.
- [GW2000a] Guarino N., Welty C. A.: Identity, Unity, and Individuality: Towards a formal toolkit for ontological analysis. W: Proceedings of ECAI-2000: The European Conference on Artificial Intelligence. Berlin, IOS Press, 2000, s. 219-223.
- [GW2000b] Guarino N., Welty C. A.: A Formal Ontology of Properties. W: Proceedings of EKAW-2000: The 12th International Conference on Knowledge Engineering and Knowledge Management. Springer Verlag, LNCS Vol. 1937, 2000, s. 97-112.
- [GW2000c] Guarino N., Welty C. A.: Ontological Analysis of Taxonomic Relationships. W: Proceedings of ER-2000: The 19th International Conference on Conceptual Modeling. Springer Verlag, LNCS Vol. 1920, 2000, s. 210-224.
- [GW2000d] Guarino N., Welty C. A.: Towards a methodology for ontology-based model engineering. W: Proceedings of ECOOP-2000 Workshop on Model Engineering. Cannes, France, 2000.
- [GW2004] Guarino N., Welty C. A.: An Overview of OntoClean. W: Staab S., Studer R. (red.): Handbook on Ontologies. Springer Verlag, Berlin, Heidelberg, 2004, s. 151-171.
- [G1992] Guha R.: Contexts: a formalization and some applications. Praca doktorska, 1992.
- [HM2001] Haarslev V., Müller R.: RACER System Description. W: Lecture Notes in Computer Science, Vol. 2083/2001, 2001, s. 701-705.
- [HS2005] Haase P.: Stojanovic L., Consistent evolution of OWL ontologies. W: Proceedings of the Second European Semantic Web Conference, Heraklion, Greece, 2005.

- [HPS2006] Halaschek-Wiener C., Parsia B., Sirin E.: Towards Incremental Reasoning Through Updates in OWL-DL. W: Proceedings of the Reasoning on the Web Workshop, WWW 2006, 2006.
- [H2004] Hartman J. (red.): Słownik filozofii, Wydawnictwo Zielona Sowa, Kraków 2004.
- [H1969] Hintikka J.: Knowledge and Belief. An Introduction to the Logic of the Two Notions, Cornell University Press, Ithaca 1962.
- [H1992] Hintikka J.: Eseje logiczno-filozoficzne, tłum. Grobler A., Wydawnictwo Naukowe PWN, W-wa 1992.
- [H1998] Horrocks I.: The Fact System. W: Proceedings of the International Conference on Automated Reasoning with Analytic Tableau and Related Methods, 1998. s. 307-312.
- [HMS2004] Hustadt U., Motik B., Sattler U.: Reducing *SHIQ* Description Logic to Disjunctive Datalog Programs. W: Proc. of the 9th International Conference on Knowledge Representation and Reasoning (KR2004), 2004, s. 152-162.
- [J1983] John-Laird P. N.: Mental Models. Harvard University Press, Cambridge, Massachusetts, 1983.
- [K1978] Kaplan D.: On the Logic of Demonstratives. W: Journal of Philosophical Logic nr 8, 1978, s. 81-98.
- [K2000] Kopaliński W.: Słownik wyrazów obcych. Świat Książki, Warszawa, 2000.
- [KM1979] Kowalski S., Mostowski A. W.: Teoria automatów i lingwistyka matematyczna. PWN, Warszawa, 1979.
- [KLWZ2003] Kutz O., Lutz C., Wolter F.: Zakharyashev M., E-connections of Description Logics. W: Proceedings of the International Workshop on Description Logics, DL'03, Rzym, 2003, s. 178–187.
- [KLWZ2004] Kutz O., Lutz C., Wolter F.: Zakharyashev M., E-connections of abstract description systems. W: Artificial Intelligence, Vol. 156, Czerwiec. 2004, s. 1-73.
- [KWZ2002] Kutz O., Wolter F.: Zakharyashev M., Connecting abstract description systems. W: Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning 2002, Morgan Kaufmann, 2002, s 215-226.
- [LN1984] Lindsay P. H., Norman D. A.: Procesy przetwarzania informacji u człowieka. PWN, Warszawa, 1984.
- [L1987] Lloyd J. W.: Foundations of Logic Programming, wydanie 2. rozszerzone, Springer Verlag, Berlin, 1987.
- [MCR2006] Mascardi V., Cordi V., Rosso P.: A Comparison of Upper Ontologies. Raport Techniczny DISI-TR-06-21, Uniwersytet w Genui, 2006.
- [M1993] McCarthy J.: Notes on Formalizing Context. W: Proceedings of IJCAI 1993, Morgan Kaufmann, 1993, s. 562-555.
- [MH1969] McCarthy J., Hayes P. J.: Some Philosophical Problems from the Standpoint of AI. W: Machine Intelligence 4/1969, Edinburgh University Press, Edinburgh, s. 463-502.

- [M1975] Minsky M.: A framework for representing knowledge. W: Brachman R. J. i Levesque H. J. (red): Readings in Knowledge Representations. Morgan Kaufmann, San Mateo, California, 1985, s. 245-262.
- [MW2004] Mitra P., Wiederhold G.: An Ontology-Composition Algebra. W: Handbook on Ontologies, Springer Verlag, 2004, s. 171-216.
- [MSH2007] Motik B., Shearer R., Horrocks I.: A Hypertableau Calculus for SHIQ. W: Proceedings. of the 20th Int. Workshop on Description Logics DL'07. Brixen/Bressanone, Włochy, 2007, s. 419–426.
- [M1998] Mulawka J. J.: Systemy ekspertowe. WNT, Warszawa 1998.
- [NP2001] Niles I., Pease A.: Towards a standard upper ontology. W: Welty C., Smith B. (red): Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). ACM Press, 2001, s 2-9.
- [N2004] Noy N. F.: Tools for Mapping and Merging Ontologies. W: Staab S. (ed.), Studer R. (ed.): Handbook on Ontologies. Springer Verlag, Berlin. Heidelberg, 2004, s. 365-384.
- [NR2006] Noy N. (red.), Rector A. L. (red.): Defining n-ary relations on the SemanticWeb. Working group note, W3C, 12.04.2006. Dostępny pod adresem: <http://www.w3.org/TR/swbp-n-aryRelations/>.
- [Q1961] Quine W.: On what there is. W: From a logical point of view, Harvard University Press, Cambridge, USA, 1961, s. 1-19.
- [Q1966] Quilan M. R.: Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities. W Brachman R. J. i Levesque H. J. (red): Readings in Knowledge Representations. Morgan Kaufmann, San Mateo, California, 1985, s. 97-118.
- [R2003] Rector A. L.: Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. W: K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture, Nowy York, USA, ACM Press, 2003, s. 121-128.
- [R1965] Robinson J. A.: A machine-oriented logic based on the resolution principle. W: Journal of the ACM (JACM), Wol. 12/1, 1965, s. 23-41.
- [RR1996] Rogers J., Rector A.: The GALEN ontology. W: Medical Informatics Europe. Amsterdam, IOS Press, 1996, s. 174-178.
- [SPOP2004] Semy S. K., Pulvermacher M. K., Obrst L. J., Pulvermacher M. K.: Toward the use of an upper ontology for U.S. government and U.S. military domains: An evaluation. Raport MTR 04B0000063 Mitre Corporation, 2004.
- [ST2005a] Serafini L., Taminin A.: Distributed instance retrieval in heterogeneous ontologies. W: Proc. of the Second Italian Semantic Web Workshop Semantic Web Applications and Perspectives (SWAP'05), 2005.
- [ST2005b] Serafini L., Taminin A.: DRAGO: Distributed Reasoning Architecture for the Semantic Web. In The Semantic Web: Research and Applications, 2005, s. 361-376.

- [ST2007] Serafini L., Tamin A.: Instance Migration in Heterogeneous Ontology Environments. W: The Semantic Web, LNCS, Vol. 4825, ISBN Springer Verlag, Berlin Heidelberg, 2007, s. 452-465.
- [SR2006] Seidenberg J., Rector A. L.: Web Ontology Segmentation: Analysis, Classification and Use. W: Proceedings of the World Wide Web Conference (WWW), Edinburgh, 2006.
- [S1987] Simons P.: Parts: A Study on Ontology. Clarendon Press, Oxford, 1987.
- [SPGKK2007] Sirin E., Parsia B., Grau B. C., Kalyanpur A., Katz Y. Pellet: A practical OWL-DL reasoner. W: Journal of Web Semantics: Science, Services and Agents on the World Wide Web 5 (2007), s. 51-53.
- [S1995] Sowa, J. F.: Top-level ontological categories. W: International Journal of Human and Computer Studies, No 43. 1995, s. 669-685.
- [S2000] Sowa J. F.: Knowledge Representation. Logical, Philosophical and Computational Foundations. Brooks/Cole Thomson Learning, Pacific Grove, California, 2000.
- [SK2004] Stuckenschmidt H., Klein M. C. A.: Structure-based partitioning of large concept hierarchies. W: International Semantic Web Conference, 2004, s. 289-303.
- [S1978] Szymczak M. (red.): Słownik języka polskiego. PWN, Warszawa 1978.
- [T1937] Tarski A.: Über den Begriff der logischen Folgerung. W: Actes du Congrès International de Philosophie Scientifique, Vol. VII, Paris, 1937, s. 1-11.
- [T2001] Tobies S.: Complexity Results and Practical Algorithms for Logics in Knowledge Representation. Praca doktorska, 2001.
- [TH2006] Tsarkov D., Horrocks I.: FaCT++ Description Logic Reasoner: System Description. W: LNCS, Vol. 4130/2006, 2006, s. 292-297.
- [TSB2006] Turhan A., Springer T., Berger M.: Pushing Doors for Modeling Contexts with OWL DL a Case Study. W: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops. IEEE Computer Society, 2006, s. 13.
- [U1988] Ullman J. D.: Systemy baz danych. WNT, Warszawa, 1988.
- [W2007] Waloszek W.: Metody strukturalnej analizy ontologii opartych na logice opisowej. Praca doktorska, 2007.
- [W2003] Wójcicki R.: Wykłady z logiki z elementami teorii wiedzy. Wydawnictwo Naukowe „Scholar”, Warszawa, 2003.
- [Z2008] Zawadzka T.: Integracja heterogenicznych źródeł wiedzy z wykorzystaniem logiki opisowej. Praca doktorska, 2008.
- [Z1996] Ziemiński Z., Logika praktyczna. PWN, Warszawa, 1996.

Wykaz definicji

Definicja 2.1 (konceptualizacja).....	18
Definicja 2.2 (zamierzona struktura świata)	18
Definicja 2.3 (przyporządkowanie ontologiczne)	18
Definicja 2.4 (zbiór zamierzonych modeli).....	18
Definicja 2.5 (konstruktory konceptów formalizmu <i>SHOIQ</i>)	21
Definicja 2.6 (konstruktory ról formalizmu <i>SHOIQ</i>)	22
Definicja 2.7 (aksjomaty i asercje formalizmu <i>SHOIQ</i>).....	22
Definicja 2.8 (ontologia dla formalizmu \mathcal{L}).....	23
Definicja 2.9 (interpretacja dla formalizmu <i>SHOIQ</i>)	23
Definicja 2.10 (spełnianie i model <i>SHOIQ</i>).....	24
Definicja 3.1 (właściwość immanentna, nieimmanentna i przeciwimmanentna).....	30
Definicja 3.2 (kryterium tożsamości).....	31
Definicja 3.3 (właściwość przenosząca kryterium tożsamości).....	31
Definicja 3.4 (właściwość dostarczająca kryterium tożsamości).....	31
Definicja 3.5 (sortal)	32
Definicja 3.6 (całość)	32
Definicja 3.7 (właściwość przenosząca całość)	32
Definicja 3.8 (przeciwcałość).....	33
Definicja 3.9 (właściwość zewnętrznie zależna).....	33
Definicja 4.1 (model PLC)	53
Definicja 4.2 (spełnianie, spełnialność i prawdziwość)	54
Definicja 4.3 (wyprowadzalność)	54
Definicja 4.4 (łańcuch kompatybilności)	60
Definicja 4.5 (relacja kompatybilności i model LMS).....	60
Definicja 4.6 (spełnianie i wynikanie)	61
Definicja 4.7 (reguła pomostowa).....	61
Definicja 4.8 (system wielokontekstowy (MCS)).....	62
Definicja 4.9 (model (DFOL))	64
Definicja 4.10 (reguły pomostowe).....	65
Definicja 4.11 (reguły odpowiedniości osobników)	66
Definicja 4.12 (rozproszona baza wiedzy (DDL))	66
Definicja 4.13 (rozproszona interpretacja).....	67
Definicja 4.14 (d-spełnialność)	67
Definicja 4.15 (połączony Lbox).....	70
Definicja 4.16 (definicje konceptów).....	70
Definicja 4.17 (dozwolone asercje).....	70
Definicja 4.18 (połączona baza wiedzy)	70
Definicja 4.19 (połączona interpretacja)	71
Definicja 4.20 (spełnianie w połączonej bazie wiedzy).....	71
Definicja 5.1 (struktura kontekstowej bazy wiedzy).....	75
Definicja 5.2 (funkcja zawartości struktury kontekstowej bazy wiedzy)	76
Definicja 5.3 (kontekstowa baza wiedzy)	77
Definicja 5.4 (agregacja bezpośrednia, dziedziczenie bezpośrednie)	77
Definicja 5.5 (ścieżka dziedziczenia).....	77
Definicja 5.6 (kontekst).....	78
Definicja 5.7 (sygnatura typu i wystąpienia kontekstu).....	79
Definicja 5.8 (sygnatura kontekstu)	79
Definicja 5.9 (interpretacja kontekstowa)	80
Definicja 5.10 (spełnianie)	81

Definicja 5.11 (wynikanie kontekstowe)	82
Definicja 6.1 (rzut regularny wystąpienia kontekstu)	107
Definicja 6.2 (schemat kontekstowej bazy wiedzy wg metody SIM).....	108
Definicja 6.3 (instancja kontekstowej bazy wiedzy wg metody SIM).....	108
Definicja 6.4 (immanentność konceptu)	115
Definicja 6.5 (immanentność lokalna względem podzbioru parametrów).....	116
Definicja 6.6 (immanentność lokalna w SIM względem podzbioru parametrów).....	117
Definicja 6.7 (immanentność lokalna ról w SIM).....	119