

POLITECHNIKA WARSZAWSKA
Wydział Elektroniki i Technik Informatycznych



ROZPRAWA DOKTORSKA

Formalna semantyka i wnioskowanie
o pojęciowym diagramie klas w UML

mgr inż. Marcin Szlenk

Promotor: prof. dr hab. inż. Krzysztof Sacha

Warszawa, wrzesień 2005

Spis treści

Przedmiot i tezy rozprawy	7
1 Wprowadzenie	8
1.1 Modelowanie systemów informatycznych	8
1.1.1 Metody strukturalne i obiektowe	8
1.1.2 Proces wytwórczy systemu	9
1.1.3 Język modelowania	9
1.2 Zunifikowany język modelowania	10
1.2.1 Typy diagramów w UML	10
1.2.2 Mechanizmy rozszerzalności UML	10
1.2.3 Specyfikacja UML	11
1.3 Diagram klas	12
1.3.1 Perspektywy interpretacji diagramu klas	12
1.3.2 Elementy diagramu klas	13
1.3.3 Pojęciowy diagram klas	15
1.4 Formalizacja semantyki	16
1.4.1 Formalny język modelowania	17
1.5 Prace nad formalizacją diagramu klas	18
2 Notacja matematyczna	21
3 Składnia i semantyka	24
3.1 Składnia abstrakcyjna	24
3.2 Klasyfikatory	25
3.3 Liczność	25
3.4 Definicja pojęciowego diagramu klas	26
3.5 Znaczenie klasyfikatora	29
3.6 Znaczenie specjalizacji	29
3.7 Stan dziedziny	30
3.8 Relacja spełniania	31
3.9 Znaczenie diagramu	33
4 Spójność semantyczna	35
4.1 Poprawność a spójność	35
4.2 Spójność klasyfikatora	36

4.3	Abstrakcyjny liść	37
4.4	Agregacja rekursywna	38
4.5	Spójność diagramu	40
5	Wynikanie semantyczne	42
5.1	Relacja wynikania	42
5.2	Reguły wynikania	43
5.3	Reguły wynikania dla klasy	43
5.4	Reguły wynikania dla asocjacji	44
5.5	Reguły wynikania dla agregacji i kompozycji	48
5.6	Reguły wynikania dla klasy asocjacji	51
5.7	Reguły wynikania dla związku specjalizacji	52
5.8	Wynikanie a spójność semantyczna	60
5.9	Złożona agregacja rekursywna	61
6	Równoważność semantyczna	63
6.1	Relacja równoważności	63
6.2	Precyzyjność diagramu	64
6.3	Unikalność powiązań a licznosc	64
6.4	Kompozycja a licznosc	66
6.5	Specjalizacja asocjacji w metamodelu UML	69
6.6	Specjalizacja asocjacji a licznosc	69
6.7	Specjalizacja asocjacji a unikalnosc powiazan	72
6.8	Specjalizacja agregacji	74
6.9	Specjalizacja kompozycji	76
6.10	„Pusta” klasa asocjacji	77
7	Semantyka ograniczeń	79
7.1	Ograniczenie	79
7.2	Znaczenie ograniczenia	80
7.3	Diagram z ograniczeniem	80
7.4	Ograniczenia zbioru generalizacji	82
7.5	Spójność a ograniczenia	83
7.6	Wynikanie z ograniczeniami	84
8	Podsumowanie	86
	Bibliografia	88
A	Słownik terminów anglojęzycznych	94
B	Zestawienie reguł wynikania	96
C	Zestawienie wzorców niespójności	99

Spis rysunków

1.1	Elementy diagramu klas	13
1.2	Pojęciowa równoważność atrybutu i asocjacji	16
1.3	Struktura języka modelowania	17
3.1	Fragment hierarchii klasyfikatorów w metamodelu UML	25
3.2	Semantyka specjalizacji	30
4.1	Niespójna agregacja rekursywna	38
4.2	Niespójność dla agregacji rekursywnej	40
4.3	Złożona agregacja rekursywna	41
5.1	Usunięcie klasy	44
5.2	Usunięcie asocjacji	45
5.3	Rozszerzenie liczności końca asocjacji	46
5.4	Usunięcie unikalności powiązań	47
5.5	Usunięcie atrybutów	48
5.6	Oslabienie agregacji	49
5.7	Oslabienie kompozycji	50
5.8	Przekształcenie klasy asocjacji w klasę	52
5.9	Usunięcie abstrakcyjności klasyfikatora	53
5.10	Usunięcie związku specjalizacji	54
5.11	Promocja asocjacji	56
5.12	Promocja agregacji	59
5.13	Niespójność we wniosku semantycznym	61
5.14	Niespójność dla złożonej agregacji rekursywnej	62
6.1	Unikalność powiązań a liczność	65
6.2	Kompozycja a liczność	67
6.3	Nieprecyzyjna liczność końca kompozycji	68
6.4	Niespójność dla kompozycji	68
6.5	Unikalność powiązań a kompozycja	69
6.6	Specjalizacja asocjacji a liczność	70
6.7	Nieprecyzyjna specyfikacja liczności	72
6.8	Specjalizacja a unikalność powiązań	73
6.9	Specjalizacja agregacji	74
6.10	Specjalizacja kompozycji	77

6.11	Równoważność klasy asocjacji i asocjacji	78
7.1	Ograniczenie {xor}	81
7.2	Stereotyp «singleton»	81
7.3	Zbiór generalizacji	82
7.4	Niespójna specjalizacja wielokrotna	84
7.5	Wynikanie z ograniczeniami	85

Spis tabel

1.1	Diagramy struktury	11
1.2	Diagramy zachowania	11
1.3	Formalizowane zagadnienia	20
3.1	Przykładowe oznaczenia liczości w UML	26
3.2	Możliwe specjalizacje klasyfikatorów	28

Przedmiot i tezy rozprawy

Unified Modeling Language (UML) jest szeroko stosowanym, graficznym językiem obiektowego modelowania systemów informatycznych o dobrze określonej składni, lecz bez jednoznacznie zdefiniowanej semantyki. Semantyka UML — określona w standardzie tego języka — abstrahuje od procesu wytwórczego systemu i wyrażona jest w języku naturalnym. Ogranicza to możliwości ścisłego wnioskowania, w tym weryfikacji zarówno samych modeli, jak również ich przekształceń.

W skład UML wchodzi różnego typu diagramy, z których centralnym jest diagram klas, opisujący statyczną strukturę systemu w kontekście klas i relacji pomiędzy nimi. W procesie tworzenia systemu, diagram klas używany jest do zapisu wyników prac zarówno analitycznych (model pojęciowy), projektowych (model projektowy), jak i implementacyjnych (model implementacyjny). W każdym z tych zastosowań diagram różni się szczegółowością, typowymi konstrukcjami oraz interpretacją. Przedmiotem niniejszej rozprawy jest diagram używany do zapisu modeli pojęciowych, nazywany pojęciowym diagramem klas.

W świetle wcześniejszych uwag na temat semantyki UML formułuje się następujące tezy rozprawy:

- Możliwe jest formalne (matematyczne) zdefiniowanie semantyki pojęciowego diagramu klas, zapewniające jego jednoznaczność interpretacji.
- Formalna semantyka pojęciowego diagramu klas umożliwi ścisłe wnioskowanie na temat błędów semantycznych na diagramie oraz zależności semantycznych pomiędzy diagramami.

Powyższe tezy wykazane zostaną poprzez, odpowiednio:

- Definicję semantyki pojęciowego diagramu klas, obejmującą formalny opis dziedziny syntaktycznej, semantycznej oraz odwzorowanie pomiędzy tymi dziedzinami, które przypisuje poszczególnym diagramom ich znaczenie.
- Definicję — w oparciu o formalnie wyrażoną semantykę — pojęć spójności, wynikania i równoważności semantycznej pojęciowych diagramów klas wraz z analizą ich własności i przykładów zastosowania.

Rozdział 1

Wprowadzenie

Niniejszy rozdział jest przeglądem podstawowych pojęć dotyczących modelowania w inżynierii oprogramowania. Stanowi wprowadzenie do UML, a w szczególności diagramu klas oraz zagadnienia formalizacji jego semantyki.

1.1 Modelowanie systemów informatycznych

Przez *system informatyczny* (dalej nazywany *systemem*) rozumie się oprogramowanie osadzone na pewnej konfiguracji sprzętowej. W inżynierii oprogramowania, *model* jest abstrakcją tworzonego systemu, widzianą z określonej perspektywy i na określonym poziomie szczegółowości. Pojęcia i notacje używane do zapisu modeli, przyjmowane perspektywy i poziomy szczegółowości oraz kolejność ich opracowywania należą do podstawowych elementów *metody* (*metodyki*) wytwarzania systemu.

1.1.1 Metody strukturalne i obiektowe

Tworzenie modeli zawsze opiera się na dekompozycji systemu. Dwie najczęściej stosowane techniki to *podejście strukturalne* i *podejście obiektowe* [Flasiński97]. W podejściu strukturalnym, podstawowym blokiem konstrukcyjnym modelu jest funkcja; w podejściu obiektowym — obiekt i jego klasa. Uzupełnieniem dekompozycji systemu w postaci funkcji jest przepływ danych i sterowania pomiędzy nimi, a dla dekompozycji w postaci obiektów — ich wzajemne powiązania, zachowanie się oraz interakcje.

Przykładowe metody wykorzystujące podejście strukturalne (*metody strukturalne*) to *metoda Yourdona* [Yourdon96] i jej modyfikacje dla *systemów czasu rzeczywistego*: *metoda Warda-Mellora* [Ward85, Sacha96] i *metoda Hatleya-Pirbhaiego* [Hatley88]. Wśród metod bazujących na podejściu obiektowym (*metody obiektowe*) można wymienić: *metodę Shlaer-Mellora* [Shlaer88], *metodę Martina-Odel-la* [Martin91], *metodę Coad-Yourdona* [Coad91a, Coad91b], *metodę Rumbaugh* [Rumbaugh91], *metodę Jacobsona* [Jacobson92], *metodę Boocha* [Booch94] i wiele innych.

1.1.2 Proces wytwórczy systemu

Na *proces wytwórczy systemu* składa się zbiór czynności i związanych z nimi wyników, które prowadzą do powstania systemu. Chociaż istnieje wiele różnych procesów wytwórczych systemu, pewne zasadnicze czynności są wspólne dla nich wszystkich. Do czynności tych należą między innymi:

- *Analiza* — obejmuje zebranie i uporządkowanie wymagań użytkownika oraz określenie, co system będzie robił, by spełnić zidentyfikowane wymagania.
- *Projektowanie* — obejmuje odwzorowanie wymagań w konstrukcję systemu, czyli określenie, jak system będzie zrealizowany. Do prac projektowych należy dobranie architektury systemu, struktury oprogramowania, języka, środowiska i sposobów implementacji.
- *Implementacja* — polega na zbudowaniu działającego systemu zgodnie z zasadami określonymi w trakcie projektowania.

Produktem każdej z powyższych czynności jest właściwy dla niej model lub zbiór modeli. W *klasycznym (kaskadowym) cyklu* budowy systemu przyjmuje się, że czynności analizowania, projektowania i implementacji są odrębnymi etapami procesu, a każdy z nich musi zostać zakończony przed rozpoczęciem kolejnego. Typowymi dla modelowania obiektowego są procesy wytwórcze oparte o *cykl iteracyjno-przyrostowy* (np. *Rational Unified Process* (RUP) [Kroll03]), na który składa się ciąg kolejnych iteracji, z których każda obejmuje wszystkie wyżej wymienione czynności.

1.1.3 Język modelowania

Do zapisu modeli wykorzystuje się *język modelowania*. Dwa podstawowe elementy języka modelowania to jego *składnia (notacja)* i *semantyka*. Składnia języka określa, jakie oznaczenia (elementy składniowe) wolno stosować i w jaki sposób wolno je ze sobą łączyć; semantyka — jak należy interpretować przyjęte oznaczenia i tworzone z nich konstrukcje składniowe (wyrażenia języka).

W zależności od charakteru stosowanej notacji, języki modelowania dzieli się ogólnie na *języki tekstowe* i *języki graficzne* [Harel00]. Te ostatnie stały się w inżynierii oprogramowania szczególnie popularne. Dla modelowania strukturalnego, ich przykładami mogą być: *diagram przepływu danych (data flow diagram, DFD)* [DeMarco79], *diagram zmiany stanów (state transition diagram, STD)* [Ward85], czy *diagram encja-związek (entity-relationship diagram, ERD)* [Chen76]; a dla modelowania obiektowego — szereg typów diagramów nazywanych wspólnie *zuniifikowanym językiem modelowania (unified modeling language, UML)* [Rumbaugh04]. Zaletą graficznych języków modelowania jest przede wszystkim ich wysoka komunikatywność, natomiast wadą bywa często brak jednoznacznej interpretacji semantycznej.

Szczególnym przypadkiem języków modelowania są *formalne języki specyfikacji*, na których opierają się tzw. *metody formalne* [Szejko02, Sacha96]. Do takich języków należą między innymi VDM-SL [Jones90, Andrews91] oraz Z [Spivey92]. Są to tekstowe języki o formalnie zdefiniowanej składni i semantyce. Jednoznaczność semantyczna tych języków i ich matematyczny charakter pozwalają na całkowicie weryfikowalne (takie, którego poprawność może być udowodniona), stopniowe przekształcanie modelu w docelowy system.

1.2 Zunifikowany język modelowania

Duża popularność podejścia obiektowego (obiektywności) i graficznych języków modelowania doprowadziła do powstania szeregu notacji, różniących się nazwami i oznaczeniem tych samych, wspólnych dla obiektywności pojęć. Język UML powstał jako unifikacja języków graficznych używanych w metodach obiektowych. Od wersji oznaczonej numerem 1.1, język ten jest standardem zaakceptowanym przez organizację *Object Management Group* (OMG)¹.

Aktualnie² trwa proces zastępowania wersji języka 1.5 [UML03] wersją 2.0 [UML04]. Ta ostatnia została przyjęta jako referencyjna w niniejszej rozprawie. Tam, gdzie istotne będzie wskazanie zmian wprowadzonych w nowej wersji, stosuje się dla uproszczenia oznaczenie „UML2” dla wersji 2.0 i „UML1” dla wersji wcześniejszych (1.1–1.5).

1.2.1 Typy diagramów w UML

Język UML jest bardzo obszerny. Wchodzące w jego skład typy diagramów dzieli się ogólnie na dwa rodzaje:

- *diagramy struktury* — pokazują statyczną strukturę obiektów w systemie (tab. 1.1);
- *diagramy zachowania* — pokazują dynamiczne zachowanie obiektów w systemie (tab. 1.2).

Mnogość dostępnych typów diagramów pozwala na tworzenie modeli przedstawiających system z różnych perspektyw i pokazujących te jego aspekty, które są istotne na danym etapie budowy.

1.2.2 Mechanizmy rozszerzalności UML

W UML istnieje możliwość definiowania i wykorzystywania nowych elementów składniowych diagramów, a także możliwość umieszczenia na diagramach dodatkowych informacji, których nie można wyrazić przy użyciu standardowych elementów języka. Dostępne w UML *mechanizmy rozszerzalności* języka to [Płodzień03]:

¹Internet: www.omg.org

²wrzesień 2005

Tabela 1.1: Diagramy struktury

Typ diagramu	Obecność w UML1
klas (ang. <i>class diagram</i>)	+
obiektów (<i>object</i>)	nieoficjalny
pakietów (<i>package</i>)	nieoficjalny
strukturalny (<i>composite structure</i>)	–
komponentów (<i>component</i>)	+
wdrożeniowy (<i>deployment</i>)	+

Tabela 1.2: Diagramy zachowania

Typ diagramu	Obecność w UML1
przypadków użycia (ang. <i>use case diagram</i>)	+
maszyny stanowej (<i>state machine</i>)	stanów (<i>statechart</i>)
aktywności (<i>activity</i>)	+
sekwencji (<i>sequence</i>)	+
komunikacji (<i>communication</i>)	współpracy (<i>collaboration</i>)
przeglądu interakcji (<i>interaction overview</i>)	–
przebiegów czasowych (<i>timing</i>)	–

- *stereotyp* — umożliwia zdefiniowanie nowego elementu na bazie już istniejącego elementu diagramu poprzez tzw. *meta-klasyfikację*;
- *wartość etykietowana* — jest nazwaną wartością (parą nazwa-wartość) dołączoną do elementu diagramu;
- *ograniczenie* — jest warunkiem nałożonym na element diagramu zapisanym tekstowo w języku naturalnym bądź formalnym, np. we wchodzącym w skład UML języku *Object Constraint Language* (OCL) [Warmer03, OCL03];
- *komentarz* — jest adnotacją przypisaną do elementu diagramu, nie mającą wpływu na znaczenie modelu opisanego tym diagramem.

Powyższe mechanizmy umożliwiają tworzenie *profilu* [Rumbaugh04], tj. swego rodzaju dialektów UML dopasowanych do konkretnej dziedziny zastosowań lub technologii implementacji.

1.2.3 Specyfikacja UML

Na pełną specyfikację UML składają się cztery dokumenty:

- *UML 2.0 Infrastructure* [UML03a] — definiuje podstawowe pojęcia, które mogą być używane przez inne specyfikacje, w tym *UML 2.0 Superstructure*;
- *UML 2.0 Superstructure* [UML04] — definiuje UML widziany na poziomie użytkownika, tj. specyfikuje dostępne konstrukcje i diagramy;

- *UML 2.0 Object Constraint Language* [OCL03] — definiuje formalny język zapisu wyrażeń (takich jak niezmienniki, warunki wstępne i końcowe) dla modeli opisanych diagramami;
- *UML 2.0 Diagram Interchange* [UML03b] — definiuje sposób zapisu i wymiany diagramów, zachowujący dokładną informację o ich graficznej reprezentacji.

Metamodel UML

Specyfikacja konstrukcji dostępnych w UML zapisana została w postaci *metamodelu* [Clark04], czyli modelu tych konstrukcji. Metamodel UML jest sam wyrażony w UML, a konkretnie opisany diagramami klas, uzupełnionymi o ograniczenia zapisane w języku naturalnym oraz OCL. Pomijając tu aspekty definiowania języka w nim samym, metamodel UML dotyczy jedynie składni języka, natomiast semantyka przedstawionych na nim konstrukcji opisana jest w sposób całkowicie nieformalny, przy użyciu języka naturalnego. Ponieważ UML jest niezależny od metod i procesów modelowania, opis ten określa jedynie pewne ramy dopuszczalnych interpretacji.

1.3 Diagram klas

Podstawowym diagramem w UML jest *diagram klas*, będący modyfikacją wspomnianego wcześniej diagramu encja-związek. Jest on centralnym pojęciem we wszystkich znanych metodach obiektowych [Subieta99b]. Diagram klas opisuje system w postaci klas występujących w nim obiektów oraz łączących je statycznych relacji, jednak jego interpretacja może być wieloraka.

1.3.1 Perspektywy interpretacji diagramu klas

Semantyka diagramu klas i poziom szczegółowości (wykorzystywane elementy notacji) zależą od *perspektywy*, która była brana pod uwagę w trakcie jego konstruowania. Klasyfikacja perspektyw konstruowania diagramu klas jest ściśle związana z procesem wytwórczym systemu (p. 1.1.2) i obejmuje trzy przypadki [Fowler02, Larman01]:

- *Perspektywa pojęciowa* — diagram reprezentuje pojęcia ze świata rzeczywistego niezależnie od oprogramowania. Pomiędzy występującymi na nim klasami oraz klasami je implementującymi zwykle nie ma pełnej odpowiedniości. Perspektywa ta jest typowa dla prac analitycznych.
- *Perspektywa specyfikacyjna* — diagram opisuje już oprogramowanie, przy czym istotne są *interfejsy* klas (widoczne „na zewnątrz” własności), a nie ich implementacja. Perspektywa jest uwzględniana w pracach projektowych.

- *Perspektywa implementacyjna* — diagram dokładnie odzwierciedla implementację, w konkretnej technologii i języku programowania, co jest właściwe dla prac implementacyjnych.

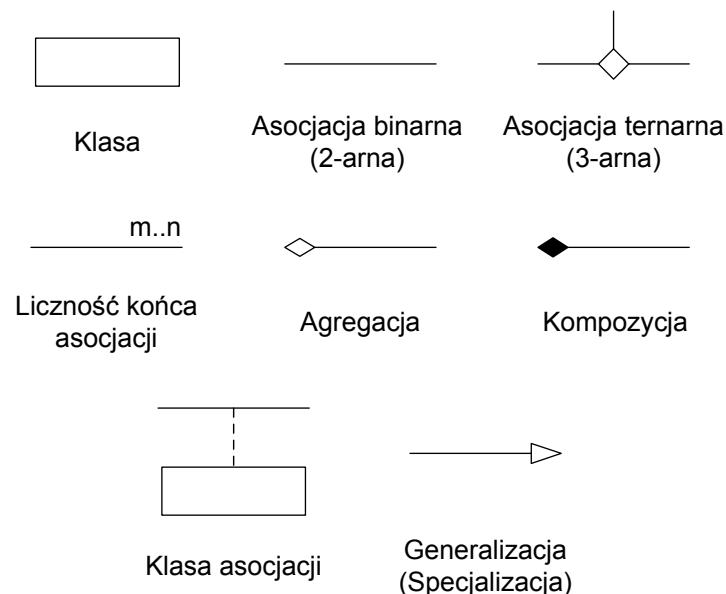
Poszczególne perspektywy mają charakter umowny, a ich granice nie są jednoznaczne. Problemy nastęrcza zwłaszcza odróżnienie perspektywy specyfikacyjnej od implementacyjnej. W ostatnich publikacjach, np. [Fowler04], zastępuje się je jedną perspektywą, nazywaną *perspektywą oprogramowania* — dla odróżnienia od perspektywy pojęciowej, jako niezależnej od oprogramowania. Sam UML nie definiuje powyższych perspektyw, pozostając językiem niezależnym od procesu modelowania.

1.3.2 Elementy diagramu klas

Do podstawowych elementów diagramu klas należą (rys. 1.1):

Klasa

Klasa jest nazwanym zbiorem obiektów o podobnych własnościach. Ogólnie, tego pojęcia używa się w kilku różnych znaczeniach [Subieta99a], z których powyższe jest najbardziej sformalizowanym. Specyfikacja klas może obejmować charakterystyczne dla obiektów *atrybuty* oraz *operacje*, jakie w kontekście obiektów mogą zostać wykonane (nie pokazane na rysunku).



Rysunek 1.1: Elementy diagramu klas

Asocjacja

Asocjacja jest związkiem pomiędzy dwoma lub więcej klasami, opisującym możliwe *powiązania* (połączenia) pomiędzy obiektami tych klas. Liczbę klas (niekoniecznie różnych) biorących udział w asocjacji nazywa się jej *arnością*, a asocjację o arności n nazywa się *n-arną*. Ograniczenie liczby obiektów innej klasy, jaka może tworzyć powiązania z jednym obiektem danej klasy definiuje się w postaci *liczności końca asocjacji*. Powiązania dla danej asocjacji są unikalne ze względu na łączone obiekty, chyba że asocjacja posiada własność $\{bag\}$ ³. Z perspektywy implementacyjnej, asocjację traktuje się jako wskaźniki lub referencje prowadzące od obiektów do obiektów.

Agregacja

Agregacja jest szczególnym przypadkiem asocjacji binarnej (2-arnej), który odwzorowuje stosunek całość-część pomiędzy obiektami. W przypadku ciągu powiązań całość-część (niekoniecznie tej samej agregacji), dla danego obiektu można wyróżnić jego *części bezpośrednio* i *pośrednio*. Klasę reprezentującą całość (na końcu oznaczonym rombem), jak i jej obiekty złożone nazywa się *agregatami*. W porównaniu do zwykłej asocjacji, agregacja nie pozwala, by ciągi powiązań tworzyły cykle. Obiekt nie może być zatem swoją bezpośrednią lub pośrednią częścią. Silniejszy warunek nakłada agregacja zwana *kompozycją*, dla której dodatkowo obiekt może być bezpośrednią częścią co najwyżej jednego agregatu, nazywanego tu *kompozytem*⁴. Słabszą formę agregacji oznacza się rombem pustym; silniejszą (kompozycję) oznacza się rombem wypełnionym.

Klasa asocjacji

Klasa asocjacji jest asocjacją, która jest również klasą i odwrotnie — klasą, która jest również asocjacją. Mimo notacji złożonej z dwóch oznaczeń jest to jeden i ten sam element modelu. Jako asocjacja, klasa asocjacji jest związkiem pomiędzy klasami; jako klasa, może zawierać atrybuty i operacje oraz sama brać udział w asocjacjach.

Generalizacja (Specjalizacja)

Generalizacja i specjalizacja to dwa określenia tego samego związku taksonomicznego pomiędzy elementem bardziej ogólnym i mniej ogólnym (specjalizacją). Na diagramie klas specjalizowane mogą być zarówno klasy, jak i asocjacje. W przypadku klas, element bardziej ogólny nosi nazwę *nadklasy*, a jego specjalizacja —

³Oznaczenie $\{bag\}$ (pol. *wielozbiór*) umieszcza się przy końcu asocjacji. W UML1 było ono traktowane jako niestandardowe ograniczenie, natomiast w UML2 wchodzi w skład standardu i jest własnością liczności końca asocjacji. W obu przypadkach dotyczy jednak semantycznie całej asocjacji [OMG-FTF, Płódzień03], dlatego dalej uważa się je za własność asocjacji.

⁴Natomiast może być pośrednią częścią wielu kompozytów [Rumbaugh04].

podklasy. Związek specjalizacji tworzy w zbiorach klas i asocjacji *hierarchię specjalizacji*, w której dla danego elementu można wyróżnić jego *specjalizacje bezpośrednie* i *pośrednie*. Z perspektywy projektowej, związek specjalizacji interpretuje się w kontekście *zgodności* interfejsów klas, a z perspektywy implementacyjnej — *dziedziczenia* atrybutów i *metod* (implementacji operacji) [Fowler02].

Ze związkiem specjalizacji związane jest pojęcie klasy i asocjacji⁵ *abstrakcyjnej*, tj. wymagającej doprecyzowania w postaci swojej specjalizacji — odpowiednio klasy lub asocjacji *konkretnej* (nie abstrakcyjnej). Nazwę klasy lub asocjacji abstrakcyjnej zapisuje się na diagramie kursywą.

1.3.3 Pojęciowy diagram klas

Diagram klas konstruowany z perspektywy pojęciowej nazywa się *pojęciowym diagramem klas*. Pojęciowy diagram klas opisuje najistotniejsze pojęcia i związki w *dziedzinie problemu* (nazywanej też dalej krótko *dziedziną*), tj. we fragmencie świata rzeczywistego, będącego przedmiotem analizy [Martin97]. Pojęcia z danej dziedziny reprezentowane są w postaci klas, nazywanych także *klasami pojęciowymi*. Klasycznym już sposobem wyszukiwania pojęć jest wyodrębnianie rzeczowników (lub fraz rzeczownikowych) w tekstowym opisie dziedziny. Przy konstruowaniu pojęciowych diagramów klas możliwe jest również wykorzystanie gotowych *wzorców analizy* [Fowler97]. Opisany pojęciowym diagramem klas model nosi nazwę *modelu dziedziny* lub *modelu pojęciowego*. Dalszą charakterystykę pojęciowego diagramu klas podaje się tu za pozycją [Larman01].

Związki

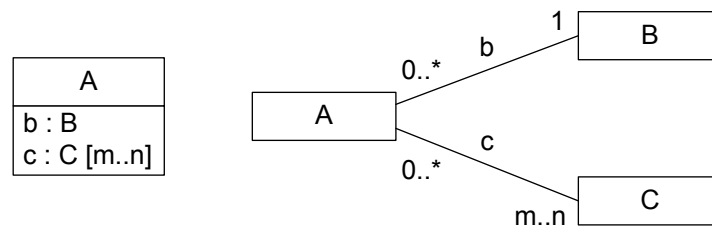
Pojęciowy diagram klas może zawierać związki asocjacji (i jej szczególne przypadki) oraz specjalizacji. Choć notacja UML umożliwia określenie kierunku asocjacji (*asocjacja skierowana*), jest to jednak czynność właściwa dopiero perspektywie specyfikacyjnej. Na pojęciowym diagramie klas nie umieszcza się pozostałych, nie wspomnianych wcześniej, związków *zależności* i *realizacji* (przedstawionych bliżej np. w [Booch01]).

Atrybuty

Jeżeli z danym pojęciem związane są atrybuty, mogą być pokazane na pojęciowym diagramie klas w postaci atrybutów klasy, a w przypadku, gdy charakteryzują asocjację — jako atrybuty klasy asocjacji. Atrybut może mieć określony *typ*, który powinien jednak reprezentować pojęcie z rzeczywistości (być klasą pojęciową). W modelu pojęciowym nie określa się *widoczności* atrybutów (tj. publiczny, chroniony, prywatny lub pakietowy [Rumbaugh04]).

Należy tutaj podkreślić, że z pojęciowego punktu widzenia nie ma żadnej różnicy pomiędzy atrybutem, a asocjacją. Równoważność atrybutu i asocjacji w modelu

⁵Abstrakcyjne asocjacje wprowadzono w UML2.



Rysunek 1.2: Pojęciowa równoważność atrybutu i asocjacji

pojęciowym pokazuje rysunek 1.2. Różnice, dotyczące przypadku użycia, a nie semantyki⁶, pojawiają się dopiero na poziomie specyfikacyjnym i implementacyjnym [Fowler04].

Operacje

Na pojęciowym diagramie klas nie specyfikuje się operacji. Operacje mogą być natomiast umownie wykorzystane np. do zapisywania *zobowiązań* klasy przy tworzeniu modeli techniką kart CRC (ang. *class-responsibility-collaboration*) [Fowler02].

1.4 Formalizacja semantyki

Formalizacja nie jest warunkiem koniecznym, jednakże wystarczającym do zagwarantowania precyzyjności języka, tj. jednoznacznej interpretacji jego wyrażeń. Istnieje wiele sposobów formalnego opisywania semantyki języków. Dotyczą one zarówno języków modelowania, jak i programowania (wyrażeniem języka modelowania jest model, a języka programowania — program). Do najczęściej wymienianych formalnych postaci semantyki [Clark04, Schmidt86, Winskel93] należą:

- *Semantyka operacyjna* — w postaci reguł zmian stanów opisuje się w jaki sposób model, bądź program wyrażony w danym języku, może być wykonywany. Znaczeniem modelu lub programu jest ciąg kolejnych stanów osiągniętych w trakcie jego wykonywania.
- *Semantyka aksjomatyczna* — język definiuje się w postaci aksjomatów i reguł wnioskowania. Nie definiuje się znaczenia modelu, bądź programu wyrażonego w danym języku, ale jego własności, które mogą być udowodnione przy użyciu wspomnianych aksjomatów i reguł.
- *Semantyka denotacyjna* — wyrażenie języka odwzorowuje się bezpośrednio w jego znaczenie, którym zazwyczaj jest obiekt matematyczny, jak np. zbiór lub funkcja. Tak zdefiniowana semantyka jest bardziej abstrakcyjna niż semantyka operacyjna (nie określa kroków wykonywania wyrażeń) i bardziej konkretna od aksjomatycznej (podaje otwarcie znaczenie wyrażeń).

⁶W UML2 wprowadzono ogólną unifikację atrybutów i asocjacji (dokładnie końców asocjacji).

- *Semantyka translacyjna* — opisuje się w jaki sposób przekształcić wyrażenia jednego języka w wyrażenia innego języka, który posiada już precyzyjną semantykę. Typowym przykładem użycia semantyki translacyjnej są kompilatory.

W niniejszej rozprawie do formalizacji semantyki pojęciowego diagramu klas wykorzystano ogólne podejście do formalizacji semantyki języków modelowania przedstawione w pracach [Harel00, Rumpe98]. Podejście to, mające wiele wspólnego z semantyką denotacyjną, prezentuje kolejny punkt.

1.4.1 Formalny język modelowania

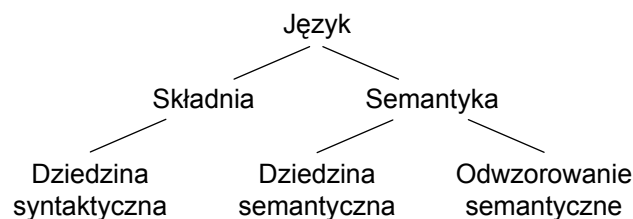
Składnia języka modelowania określa zbiór jego możliwych wyrażeń nazywany *dziedziną syntaktyczną*. Semantyka języka składa się natomiast z dwóch elementów: zbioru znaczeń wyrażeń nazywanego *dziedziną semantyczną* oraz funkcji, która przypisuje wyrażeniom ich znaczenia. Funkcja ta nosi nazwę *odwzorowania semantycznego*. Powyższą strukturę przedstawia rysunek 1.3.

Jeśli dla języka dana jest jego dziedzina syntaktyczna Syn , wówczas formalizacja semantyki polega na określeniu dziedziny semantycznej Sem oraz zdefiniowaniu odwzorowania semantycznego:

$$\mathcal{M}: Syn \rightarrow Sem. \tag{1.1}$$

Definicje funkcji \mathcal{M} oraz zbiorów Syn i Sem wymagają same użycia pewnego języka, w którym zostaną wyrażone (tzw. *metajęzyka*). Język ten musi się już charakteryzować precyzyjnie określoną semantyką. W ogólnym przypadku do opisu każdego ze wspomnianych elementów może zostać użyty odrębny język, dobrany stosownie do własności, które mają być nim wyrażone.

W niniejszej rozprawie definicję dziedziny syntaktycznej, semantycznej oraz odwzorowania semantycznego dla pojęciowego diagramu klas oparto o standardową notację matematyczną wykorzystywaną w logice i teorii mnogości [Kuratowski80, Trzęsicki03]. Wybór tej notacji, pomijając adekwatność do wyrażanych własności (co wyniknie z dalszej części rozprawy), podyktowany był jej uniwersalnością i powszechnością. Przykłady innych języków używanych do formalizacji semantyki diagramów klas podane zostały w punkcie 1.5.



Rysunek 1.3: Struktura języka modelowania

1.5 Prace nad formalizacją diagramu klas

Brak jednoznacznej semantyki UML nie tylko może prowadzić do nieporozumień i tworzenia błędnych modeli, ale także stanowi przeszkodę dla precyzyjnego wnioskowania o ich semantycznej spójności i poprawności przekształceń. Dlatego też od chwili opublikowania UML zaczęły powstawać opracowania naukowe na temat jego formalizacji. W szczególności, część prac skupiona została wokół grupy *pUML*⁷, która powstała w celu uczynienia z UML precyzyjnego języka modelowania [Evans98a, Evans98b]. Do wyników prac tej grupy w zakresie formalizacji diagramu klas należą pozycje [Evans98c] oraz [France99].

[Evans98c]

Pierwsza z nich skupia się przede wszystkim na wnioskowaniu o diagramie klas. Prezentuje reguły przekształceń, przy użyciu których z danego diagramu mogą być wywnioskowane dodatkowe własności. W pracy zdefiniowano formalnie dziedzinę syntaktyczną i semantyczną diagramu klas przy użyciu notacji *Z*, przy czym brakujące odwzorowanie semantycznie może być łatwo dedefiniowane. Formalizacja obejmuje jedynie podstawowe elementy diagramu: klasę, asocjację binarną i specjalizację klas. Dla reguł przekształceń diagramów brakuje w pracy formalnych założeń, definiujących przypadki, kiedy dana reguła może być stosowana. Założenia takie są niezbędne do ich dowodzenia.

[France99]

Druga ze wspomnianych prac nie definiuje dziedziny syntaktycznej i semantycznej diagramu klas, lecz pokazuje na przykładzie konkretnych konstrukcji graficznych, w jaki sposób zapisać ich semantykę w języku *Z* (niekompletna semantyka translacyjna). W pracy rozważa się również pewną relację wynikania diagramu z innego diagramu. Jednak w efekcie braku w pełni formalnej definicji semantyki diagramu klas, relacja ta nie mogła być zdefiniowana, a jest jedynie rozumiana intuicyjnie. Powyższe niedoskonałości powielają również wstępne prace autora niniejszej rozprawy, w których do opisu semantyki diagramu klas wykorzystywano język VDM-SL [Szlenk02, Szlenk03].

[Aredo99]

Do prac nad formalizacją diagramu klas prowadzonych poza grupą *pUML* należy pozycja [Aredo99]. Proponuje się w niej zapisywanie diagramu klas w postaci specyfikacji w języku PVS-SL [Crow95]. Sposób generowania takiej specyfikacji jest przedstawiony w pracy na przykładzie konkretnych diagramów. Postać tworzonej specyfikacji jest bliska bezpośredniemu przekształcaniu diagramów w język programowania (perspektywa implementacyjna), przy czym autorzy nie deklarują przyjętej perspektywy interpretacji. Zaletą języka PVS-SL jest możliwość analizy

⁷ *The Precise UML Group*; internet: www.cs.york.ac.uk/puml/

specyfikacji przy użyciu systemu PVS, jednak praca nie porusza wątku wnioskowania na temat diagramu klas.

[Cali02, Berardi03]

Wnioskowanie o diagramie klas jest natomiast głównym tematem prac [Cali02, Berardi03]. Diagram klas zapisuje się tu w postaci wyrażeń logik z rodziny logik opisu (ang. *description logics*). Silną stroną tego podejścia jest wsparcie w postaci systemów wnioskowania w takich logikach. Sama semantyka diagramu klas znalazła się w tych pracach niejako na drugim planie. Rozpatruje się tu przykładowe konstrukcje, nie definiując dziedziny syntaktycznej diagramu klas. Niektóre elementy definiuje się niekompletnie lub abstrahując od specyfikacji UML: dla asocjacji o arności większej niż dwa pomija się interpretację licznosci jej końców, a agregację definiuje się jako zwykłą asocjację, która — co nie ma uzasadnienia w specyfikacji UML — nie może być klasą asocjacji.

[Funes03]

Praca [Funes03] definiuje składnię, a następnie semantykę diagramu klas w języku specyfikacji metody RAISE [George92]. Uwzględnia przy tym elementy diagramu używane zarówno w perspektywie pojęciowej, jak i specyfikacyjnej. Czynność translacji diagramu klas na odpowiednią specyfikację systematyzuje się, definiując wzorce generowania fragmentów specyfikacji dla poszczególnych konstrukcji diagramu. Przedstawiony opis formalny diagramu klas charakteryzują jednak pewne ograniczenia. Klasę asocjacji oraz asocjację o arności większej niż dwa przedstawia się w postaci dekompozycji na asocjacje binarne. W drugim przypadku prowadzi to do niezgodnej ze specyfikacją UML interpretacji licznosci końców takiej asocjacji. Formalizacja nie umożliwia ponadto reprezentowania specjalizacji wielokrotnych (klasa może mieć tylko jedną bezpośrednią nadklasę). Praca pomija temat zastosowania przedstawionej formalizacji do wnioskowania o diagramie klas.

Obok prac formalizujących semantykę diagramu klas, powstają prace przedstawiające formalną interpretację wybranych elementów diagramu, takich jak związki pomiędzy klasami [Övergaard98, Hnatkowska04], w szczególności związki agregacji i kompozycji [Bruel01, Barbier03]. Warto tutaj zauważyć, że semantyka agregacji zabrania istnienia cykli wśród powiązań należących do jakichkolwiek agregacji na diagramie (niekoniecznie tej samej) [Rumbaugh04]. Nie da się wyrazić tego warunku rozważając tylko przypadek pojedynczej agregacji. Podobnie, dla związku kompozycji nie sposób w pełni wyrazić warunku, iż obiekt nie może być współdzielony przez kilka kompozytów, bez uwzględnienia wszystkich kompozycji na diagramie. Znaczenia diagramu nie da się zatem odtworzyć po jego dekompozycji na odrębnie interpretowane elementy.

Wymienione wyżej pozycje stanowią pewną próbkę prac nad formalizacją diagramu klas. W ogólnym przypadku, prace te są trudne do porównania. Różnice przejawiają się w przyjmowanych perspektywach interpretacji, bądź też ich braku,

Tabela 1.3: Formalizowane zagadnienia

Zagadnienie	Pozycje bibliograficzne
klasa/asocjacja binarna/ specjalizacja klasy	[Övergaard98], [Evans98c], [France99], [Aredo99], [Cali02, Berardi03], [Funes03], [Hnatkowska04]
asocjacja n -arna	[Cali02, Berardi03] [†] , [Hnatkowska04]
klasa asocjacji	[Cali02, Berardi03], [Hnatkowska04]
agregacja/kompozycja	[France99] ^{††} , [Aredo99] ^{††} , [Brue101] ^{††} , [Cali02, Berardi03] ^{††} , [Barbier03], [Hnatkowska04] ^{††}
specjalizacja asocjacji	–
nieunikalność powiązań	–
spójność semantyczna	[Cali02, Berardi03]
wynikanie semantyczne	[Evans98c], [France99], [Cali02, Berardi03]
równoważność semantyczna	–

a także w samej metodzie formalizacji. W wielu przypadkach ma ona charakter półformalny. Autorzy nie definiują zbioru wyrażeń języka, który formalizują, podając interpretację przykładowych diagramów.

Prace różnią się poruszonymi zagadnieniami (tab. 1.3). Najczęściej obejmują podstawowe elementy diagramu, jak: klasa, asocjacja binarna i specjalizacja klasy. Przegląd dotychczasowych prac wskazuje na całkowity brak formalnych opracowań na temat specjalizacji asocjacji i asocjacji o nieunikalnych powiązaniach. W zakresie wnioskowania o diagramie klas brakuje natomiast formalnego ujęcia zagadnienia równoważności semantycznej diagramów. Wśród nieformalnych podejść, definiujących nieweryfikowalne przekształcenia diagramów, można tu wymienić pozycje [Gogolla98, Egyed00, Egyed02].

Głównie ze względów chronologicznych, dotychczasowe prace w dziedzinie formalizacji diagramu klas nie odzwierciedlają zmian wprowadzonych w UML2.

[†]Formalizacja pomija interpretację liczności końców asocjacji.

^{††}Formalizacja traktuje agregację jako semantycznie równoważną asocjacji lub obejmuje tylko przypadek pojedynczej agregacji na diagramie. Nie formalizuje się kompozycji lub ogranicza się do przypadku, gdy na diagramie występuje tylko jeden taki związek.

Rozdział 2

Notacja matematyczna

Rozdział ten przedstawia podstawowe elementy notacji matematycznej wykorzystane w dalszej części rozprawy jako język, w którym definiuje się semantykę pojęciowego diagramu klas. Celem tego rozdziału jest ustalenie oznaczeń oraz terminologii używanej na poziomie metajęzyka. Przyjęte definicje bazują częściowo na pozycji [ISO-VDM].

Logika

Definicja semantyki pojęciowego diagramu klas wykorzystuje klasyczną logikę dwuwartościową, gdzie: T oznacza prawdę, F — fałsz, \vee — sumę logiczną, \wedge — iloczyn logiczny (koniunkcję), \Rightarrow — implikację, \Leftrightarrow — równoważność i \neg — negację.

Zbiór

Symbol \mathbb{N} oznacza zbiór liczb naturalnych, \mathbb{B} — zbiór wartości boolowskich: prawda (T) i fałsz (F). Zbiór pusty oznacza się przez \emptyset . Dla dowolnego zbioru A , $\mathcal{P}(A)$ oznacza jego *zbiór potęgowy*, czyli zbiór wszystkich podzbiorów A . Jeśli A jest zbiorem skończonym, to przez $|A|$ oznacza się *moc* zbioru A , tj. liczbę jego elementów.

Operatory na zbiorach: \cup , \cap , \bigcup , \bigcap , \setminus (suma, przecięcie, suma uogólniona, przecięcie uogólnione i różnica zbiorów) mają swoje tradycyjne znaczenie. Dla zbiorów A i B , operator \times tworzy ich *iloczyn kartezjański*, który definiuje się:

$$A \times B = \{ (a, b) : a \in A, b \in B \}.$$

Uogólniony iloczyn kartezjański n zbiorów A_1, \dots, A_n ma postać:

$$\prod_{i=1}^n A_i = \{ (a_1, \dots, a_n) : a_1 \in A_1, \dots, a_n \in A_n \}.$$

Jeżeli $A_1 = \dots = A_n = A$, to $\prod_{i=1}^n A_i$ zapisuje się A^n . Użyty w powyższych definicjach operator *krotki (n-tki) uporządkowanej* $(-, \dots, -)$ ma następującą własność:

$$\forall a_1, a'_1 \in A_1, \dots, \forall a_n, a'_n \in A_n. \\ (a_1, \dots, a_n) = (a'_1, \dots, a'_n) \Leftrightarrow a_1 = a'_1 \wedge \dots \wedge a_n = a'_n.$$

Dla iloczynu kartezjańskiego $\times_{i=1}^n A_i$ istnieje rodzina n projekcji π_1, \dots, π_n oraz projekcji $\bar{\pi}_1, \dots, \bar{\pi}_n$ takich, że dla krotki $(a_1, \dots, a_n) \in \times_{i=1}^n A_i$ i dla $k = 1, \dots, n$:

$$\begin{aligned} \pi_k((a_1, \dots, a_n)) &= a_k, \\ \bar{\pi}_k((a_1, \dots, a_n)) &= \begin{cases} (a_2, \dots, a_n), & \text{dla } k = 1 \text{ i } n \geq 2 \\ (a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_n), & \text{dla } 1 < k < n \text{ i } n \geq 2, \\ (a_1, \dots, a_{n-1}), & \text{dla } k = n \text{ i } n \geq 2 \end{cases} \end{aligned}$$

czyli wartością projekcji π_k jest k -ty element krotki, natomiast wartością projekcji $\bar{\pi}_k$ jest krotka bez elementu k -tego.

Jeżeli $x = (a_1, \dots, a_n)$, to do elementów krotki można odwoływać się również przy użyciu kwalifikatora, np. $x.a_1$.

Lista

Dla zbioru A , przez A^* oznacza się zbiór wszystkich *skończonych list* zbudowanych z elementów zbioru A :

$$A^* = \bigcup_{n \in \mathbb{N}} A^n,$$

gdzie A^0 oznacza jednoelementowy zbiór zawierający *listę pustą* ($[]$). Lista $[a_1, \dots, a_n] \in A^*$ złożona z elementów $a_1, \dots, a_n \in A$ jest formalnie równa krotce (a_1, \dots, a_n) .

Definicje rodzin projekcji π_k i $\bar{\pi}_k$ rozszerza się dla zbioru list skończonych. Dla niepustej listy $[a_1, \dots, a_n] \in A^*$ i dla $k = 1, \dots, n$:

$$\begin{aligned} \pi_k([a_1, \dots, a_n]) &= a_k, \\ \bar{\pi}_k([a_1, \dots, a_n]) &= \begin{cases} [], & \text{dla } k = 1 \text{ i } n = 1 \\ [a_2, \dots, a_n], & \text{dla } k = 1 \text{ i } n \geq 2 \\ [a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_n], & \text{dla } 1 < k < n \text{ i } n \geq 2 \\ [a_1, \dots, a_{n-1}], & \text{dla } k = n \text{ i } n \geq 2 \end{cases} \end{aligned}$$

Dla zbioru A^* definiuje się ponadto funkcję **len** zwracającą *długość* danej listy. Dla niepustej listy $[a_1, \dots, a_n] \in A^*$:

$$\mathbf{len}([a_1, \dots, a_n]) = n,$$

a dla listy pustej:

$$\mathbf{len}([]) = 0.$$

Dla zbioru A , przez $A^{*(2)}$ oznacza się zbiór list skończonych zbudowanych z co najmniej dwóch elementów zbioru A :

$$A^{*(2)} = A^* \setminus (A^0 \cup A).$$

Funkcja

Dla zbiorów A i B , przez $A \rightarrow B$ oznacza się zbiór wszystkich *funkcji częściowych* ze zbioru A w zbiór B zdefiniowany następująco:

$$A \rightarrow B = \{ f \in \mathcal{P}(A \times B) : \forall (a, b_1), (a, b_2) \in f \cdot b_1 = b_2 \}.$$

Zamiast $f \in A \rightarrow B$ używa się też zapisu $f: A \rightarrow B$. Dla funkcji $f: A \rightarrow B$ definiuje się jej *dziedzinę* $\mathbf{dom}(f)$ jako zbiór:

$$\mathbf{dom}(f) = \{ a \in A : \exists b \in B \cdot (a, b) \in f \}.$$

Dla zbiorów A i B , przez $A \rightarrow B$ oznacza się zbiór wszystkich *funkcji całkowitych* ze zbioru A w zbiór B . Jest to podzbiór $A \rightarrow B$ zdefiniowany jak niżej:

$$A \rightarrow B = \{ f \in A \rightarrow B : \mathbf{dom}(f) = A \}.$$

Rozdział 3

Składnia i semantyka

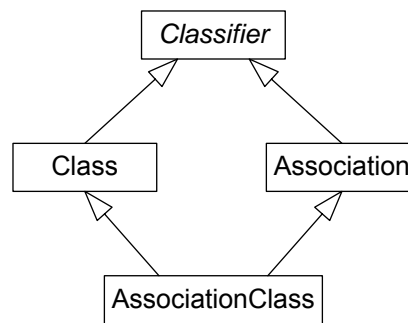
Niniejszy rozdział definiuje formalnie abstrakcyjną składnię pojęciowego diagramu klas, a następnie jego semantykę. Przedstawionym definicjom towarzyszą nieformalnie zapisane założenia oraz wyjaśnienia.

3.1 Składnia abstrakcyjna

Składnię języków graficznych opisuje się zazwyczaj na pewnym poziomie abstrakcji w postaci tzw. *składni abstrakcyjnej* (ang. *abstract syntax*). Uwzględnia się w niej istotne pojęciowe elementy notacji, pomijając ich konkretną reprezentację, jaką są różnego rodzaju linie, łuki, czy figury geometryczne. Uwzględniane dalej elementy abstrakcyjnej składni pojęciowego diagramu klas to:

- klasa:
 - konkretna/abstrakcyjna;
- asocjacja:
 - bez/z własnością {bag},
 - *zwykła*/agregacja/kompozycja,
 - konkretna/abstrakcyjna;
- klasa asocjacji;
- specjalizacja.

Nie wyszczególnia się tutaj atrybutu, który zgodnie z tym, co zostało wcześniej powiedziane, traktuje się jako inny sposób przedstawienia asocjacji. Pomija się ponadto mechanizmy rozszerzalności UML. To zagadnienie poruszone zostanie w rozdziale 7.



Rysunek 3.1: Fragment hierarchii klasyfikatorów w metamodelu UML

3.2 Klasyfikatory

Klasa, asocjacja oraz klasa asocjacji są przypadkami *klasyfikatora*⁸, czyli elementu modelu, który opisuje pewne własności strukturalne i/lub behawioralne [Rumbaugh04]. Taksonomiczną zależność pomiędzy nimi zdefiniowaną w metamodelu UML [UML04] przedstawia rysunek 3.1.

Definicja 3.2.1 (Zbiór Classifiers)

Przez Classifiers oznacza się skończony zbiór wszystkich klasyfikatorów, które mogą być elementami modelu pojęciowego, tj. klas, asocjacji i klas asocjacji, które mogą pojawić się na pojęciowym diagramie klas. □

3.3 Liczność

Asocjacja może posiadać wyspecyfikowane licznosci końców. Liczność jest zakresem nieujemnych wartości, który zapisuje się w UML jako listę przedziałów⁹, podając dla każdego z nich dolne i górne ograniczenie w postaci *minimum..maksimum*. Jeżeli ograniczenie dolne przedziału jest równe górnemu, to alternatywnie podaje się tylko ograniczenie dolne(górne). Brak ograniczenia górnego oznacza się symbolem „*”. Sam symbol „*” — bez ograniczenia dolnego — odpowiada oznaczeniu „0..*”. W skład licznosci musi wchodzić się co najmniej jedna wartość niezerowa [UML04, str. 97].

Liczność końca asocjacji może nie być wyspecyfikowana na diagramie, jednak istnieje w modelu [Rumbaugh04]. W ukończonym modelu nie można mówić o nieokreślonej licznosci i brak informacji na diagramie jest semantycznie równoważny oznaczeniu „0..*” na końcu asocjacji.

⁸UML wyszczególnia znacznie więcej typów klasyfikatora. Asocjacje zalicza się do typów klasyfikatora od UML2.

⁹UML2 nie dopuszcza licznosci nieciągłych — licznosc specyfikuje się jako jeden przedział [Fowler04]. W niniejszej pracy pozostano przy licznosciach z UML1 ze względu na możliwość bardziej precyzyjnego wyrażenia związków pomiędzy diagramami. Jest to jedyne odstępstwo od standardu UML2 w tejże pracy.

Tabela 3.1: Przykładowe oznaczenia licznosci w UML

Oznaczenie licznosci	Reprezentacja formalna
1	$\{1\}$
1..*	$\mathbb{N} \setminus \{0\}$
0..1	$\{0, 1\}$
1..3, 5	$\{1, 2, 3, 5\}$
0..*	\mathbb{N}
*	\mathbb{N}
brak	\mathbb{N}

Pojęcie licznosci reprezentowane jest formalnie jako pewien niepusty zbiór liczb naturalnych (element zbioru $\mathcal{P}(\mathbb{N}) \setminus \{\emptyset, \{0\}\}$). Związek tej reprezentacji z oznaczeniami w UML przedstawia tabela 3.1.

3.4 Definicja pojęciowego diagramu klas

Dla zbioru *Classifiers* definiuje się dalej formalnie abstrakcyjną składnię pojęciowego diagramu klas. Poniższa definicja opiera się na mechanizmie uogólniania, odzwierciedlając charakter klasy asocjacji jako szczególnego przypadku klasy i asocjacji, agregacji — jako szczególnego przypadku asocjacji, i kompozycji — jako szczególnego przypadku agregacji. Związek specjalizacji, tak jak w metamodelu UML [UML04, str. 71], zdefiniowany został pomiędzy klasyfikatorami¹⁰.

Definicja 3.4.1 (Pojęciowy diagram klas)

Pojęciowy diagram klas jest uporządkowana dziewiątką:

$$\mathcal{D} = (\text{classes, assocs, ends, mults, ASSOCS}_{\text{bag}}, \text{ASSOCS}_{\text{agg}}, \text{ASSOCS}_{\text{com}}, \text{specs, abstract}),$$

gdzie:

1. $\mathcal{D}.\text{classes}$ jest *zbiorem klas*:

$$\mathcal{D}.\text{classes} \subseteq \text{Classifiers}. \tag{3.1}$$

2. $\mathcal{D}.\text{assocs}$ jest *zbiorem asocjacji*:

$$\mathcal{D}.\text{assocs} \subseteq \text{Classifiers}. \tag{3.2}$$

W oparciu o zbiory $\mathcal{D}.\text{classes}$ i $\mathcal{D}.\text{assocs}$ definiuje się dla diagramu \mathcal{D} *zbiór klas asocjacji*:

$$\mathcal{D}.\text{asclasses} =_{\text{def}} \mathcal{D}.\text{classes} \cap \mathcal{D}.\text{assocs} \tag{3.3}$$

¹⁰Ponieważ asocjacja nie była rodzajem klasyfikatora, w metamodelu UML1 związek generalizacji/specjalizacji zdefiniowany był pomiędzy elementami pośredniego typu *GeneralizableElement*.

oraz zbiór klasyfikatorów:

$$\mathcal{D}.\text{classifiers} =_{def} \mathcal{D}.\text{classes} \cup \mathcal{D}.\text{assocs}. \quad (3.4)$$

3. $\mathcal{D}.\text{ends}$ jest *funkcją końców asocjacji*. Dla każdej asocjacji funkcja wyznacza listę co najmniej dwóch — połączonych tą asocjacją — klas:

$$\mathcal{D}.\text{ends}: \mathcal{D}.\text{assocs} \rightarrow \mathcal{D}.\text{classes}^{*(2)}. \quad (3.5)$$

Dana klasa może wystąpić na więcej niż jednym końcu asocjacji. Pozycja klasy na liście $\mathcal{D}.\text{ends}(as)$ identyfikuje koniec asocjacji, tj. *rolę* jaką klasa pełni w asocjacji.

Klasa asocjacji nie może być zdefiniowana pomiędzy samą sobą [UML04, str. 44]:

$$\begin{aligned} \forall ac \in \mathcal{D}.\text{asclasses} \cdot \forall i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(ac))\} \cdot \\ \pi_i(\mathcal{D}.\text{ends}(ac)) \neq ac. \end{aligned} \quad (3.6)$$

4. $\mathcal{D}.\text{mults}$ jest *funkcją licznosci końców asocjacji*. Dla każdej asocjacji funkcja wyznacza listę licznosci jej końców:

$$\mathcal{D}.\text{mults}: \mathcal{D}.\text{assocs} \rightarrow (\mathcal{P}(\mathbb{N}) \setminus \{\emptyset, \{0\}\})^{*(2)}. \quad (3.7)$$

Jak wyżej, pozycja licznosci na liście $\mathcal{D}.\text{mults}(as)$ identyfikuje koniec asocjacji. Arność asocjacji, która wynika z funkcji licznosci odpowiada arności wynikającej z funkcji końców:

$$\forall as \in \mathcal{D}.\text{assocs} \cdot \mathbf{len}(\mathcal{D}.\text{mults}(as)) = \mathbf{len}(\mathcal{D}.\text{ends}(as)). \quad (3.8)$$

5. $\mathcal{D}.\text{assocs}_{\text{bag}}$ jest *zbiorem asocjacji z własnością {bag}*:

$$\mathcal{D}.\text{assocs}_{\text{bag}} \subseteq \mathcal{D}.\text{assocs}. \quad (3.9)$$

6. $\mathcal{D}.\text{assocs}_{\text{agg}}$ jest *zbiorem agregacji*:

$$\mathcal{D}.\text{assocs}_{\text{agg}} \subseteq \mathcal{D}.\text{assocs}. \quad (3.10)$$

Agregacjami są tylko asocjacje binarne [UML04, str. 37]:

$$\forall as \in \mathcal{D}.\text{assocs}_{\text{agg}} \cdot \mathbf{len}(\mathcal{D}.\text{ends}(as)) = 2. \quad (3.11)$$

Przyjmuje się, że agregat jest pierwszym elementem listy $\mathcal{D}.\text{ends}(as)$.

Tabela 3.2: Możliwe specjalizacje klasyfikatorów

Klasyfikator	Specjalizacja		
	Klasa	Asocjacja	Klasa asocjacji
Klasa	+	–	+
Asocjacja	–	+	+
Klasa asocjacji	–	–	+

7. $\mathcal{D}.\text{assocs}_{\text{com}}$ jest *zbiorem kompozycji*¹¹:

$$\mathcal{D}.\text{assocs}_{\text{com}} \subseteq \mathcal{D}.\text{assocs}_{\text{agg}}. \quad (3.12)$$

8. $\mathcal{D}.\text{specs}$ jest *funkcją specjalizacji*. Dla każdego klasyfikatora funkcja wyznacza zbiór jego specjalizacji (zarówno bezpośrednich, jak i pośrednich):

$$\mathcal{D}.\text{specs}: \mathcal{D}.\text{classifiers} \rightarrow \mathcal{P}(\mathcal{D}.\text{classifiers}). \quad (3.13)$$

Domyślnie, klasyfikator może być specjalizacją klasyfikatora tego samego typu lub typu bardziej ogólnego [UML04, str. 51–52]: klasa może być specjalizacją klasy; asocjacja — asocjacji; klasa asocjacji — klasy asocjacji, klasy lub asocjacji (tab. 3.2). Formalnie, wyrażają to poniższe warunki:

$$\forall cl \in \mathcal{D}.\text{classes} \cdot \mathcal{D}.\text{specs}(cl) \subseteq \mathcal{D}.\text{classes}, \quad (3.14)$$

$$\forall as \in \mathcal{D}.\text{assocs} \cdot \mathcal{D}.\text{specs}(as) \subseteq \mathcal{D}.\text{assocs}, \quad (3.15)$$

$$\forall cf \in \mathcal{D}.\text{classifiers} \cdot \mathcal{D}.\text{specs}(cf) \not\subseteq \mathcal{D}.\text{asclasses} \Rightarrow cf \notin \mathcal{D}.\text{asclasses}. \quad (3.16)$$

Hierarchia specjalizacji jest acykliczna [UML04, str. 51], czyli żaden klasyfikator nie jest swoją specjalizacją:

$$\forall cf \in \mathcal{D}.\text{classifiers} \cdot cf \notin \mathcal{D}.\text{specs}(cf). \quad (3.17)$$

Asocjacja będąca specjalizacją innej asocjacji ma taką samą liczbę końców:

$$\begin{aligned} \forall as_1, as_2 \in \mathcal{D}.\text{assocs} \cdot as_2 \in \mathcal{D}.\text{specs}(as_1) \Rightarrow \\ \mathbf{len}(\mathcal{D}.\text{ends}(as_1)) = \mathbf{len}(\mathcal{D}.\text{ends}(as_2)), \end{aligned} \quad (3.18)$$

a na odpowiednich jej końcach znajdują się te same klasyfikatory, co na końcach asocjacji specjalizowanej lub ich specjalizacje [UML04, str. 37]:

$$\begin{aligned} \forall as_1, as_2 \in \mathcal{D}.\text{assocs} \cdot as_2 \in \mathcal{D}.\text{specs}(as_1) \Rightarrow \\ \forall i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(as_1))\} \cdot \pi_i(\mathcal{D}.\text{ends}(as_2)) \in \\ \{\pi_i(\mathcal{D}.\text{ends}(as_1))\} \cup \mathcal{D}.\text{specs}(\pi_i(\mathcal{D}.\text{ends}(as_1))). \end{aligned} \quad (3.19)$$

¹¹Specyfikacje UML1 ograniczały możliwe liczności końca kompozycji po stronie kompozytu do „1” lub „0..1”. Metamodel UML2 nie narzuca tego ograniczenia. Liczności te implikuje natomiast sama semantyka kompozycji, co zostanie pokazane w punkcie 6.4 (str. 66).

9. $\mathcal{D}.\text{abstract}$ jest zbiorem klasyfikatorów abstrakcyjnych:

$$\mathcal{D}.\text{abstract} \subseteq \mathcal{D}.\text{classifiers}. \quad \square \quad (3.20)$$

Definicja 3.4.2 (Zbiór Diagrams)

Przez Diagrams oznacza się zbiór wszystkich pojęciowych diagramów klas postaci jak w definicji 3.4.1. \square

3.5 Znaczenie klasyfikatora

Klasyfikator jest zbiorem *instancji*, które posiadają pewne wspólne własności. Instancjami klasy są *obiekty*, natomiast asocjacji — *powiązania*. W przypadku klasy asocjacji, jej instancje są jednocześnie obiektem i powiązaniem. Powiązanie łączy obiekty klas biorących udział w asocjacji. Jeżeli powiązanie jest instancją klasy asocjacji, wówczas (będąc jednocześnie obiektem) samo może tworzyć powiązania z obiektami.

Definicja 3.5.1 (Zbiór Instances)

Przez Instances oznacza się skończony¹² zbiór wszystkich potencjalnych instancji klasyfikatorów ze zbioru Classifiers. \square

Uwaga 3.5.1 (Tożsamość instancji)

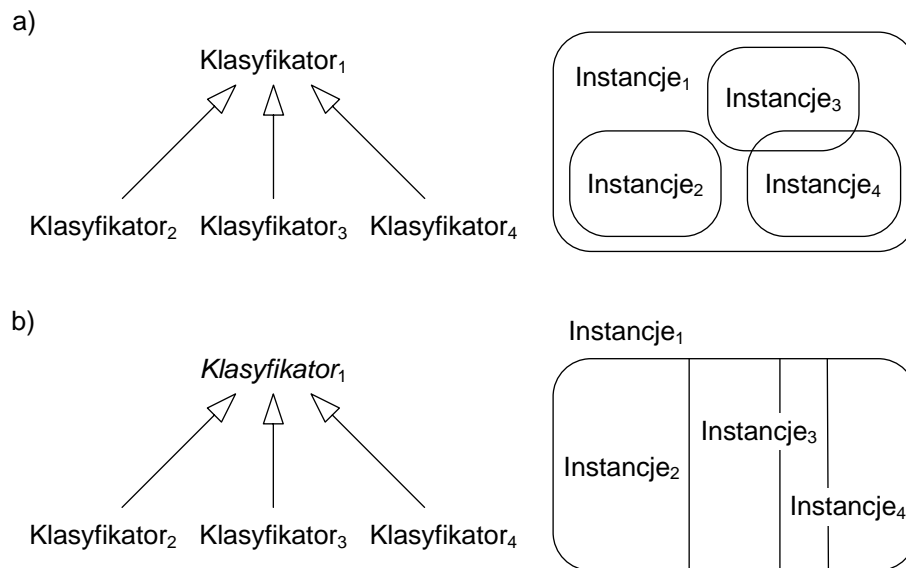
Instancje posiadają swoją *tożsamość*. W przypadku dwóch obiektów oznacza to, że są rozróżnialne nawet jeśli tworzą powiązania z dokładnie tymi samymi obiektami (mają te same wartości atrybutów). W kwestii tożsamości powiązań, specyfikacja UML nie jest konsekwentna. Powiązanie jest w niej zdefiniowane jako krotka obiektów, z czego wynika, że tożsamość powiązania wyznaczają połączone nim obiekty. W przypadku, gdy dany zbiór klas połączony jest więcej niż jedną asocjacją, lub gdy asocjacja posiada własność {bag} może jednak istnieć wiele powiązań złożonych z tych samych obiektów. Stąd, przynajmniej na poziomie pojęciowym, przyjmuje się, że powiązania posiadają tożsamość niezależną od obiektów, które łączą¹³. \square

3.6 Znaczenie specjalizacji

Związek specjalizacji klasyfikatorów jest interpretowany na poziomie pojęciowym w terminach zawierania się zbiorów instancji [Larman01]. Zbiór instancji klasyfikatora specjalizującego (*potomka* w hierarchii specjalizacji) jest podzbiorem

¹²Ponieważ systemy informatyczne są skończone stanowe, nie można przy ich użyciu reprezentować nieskończonej liczby instancji.

¹³Na problem tożsamości powiązań w UML wskazuje między innymi pozycja [Stevens02].



Rysunek 3.2: Semantyka specjalizacji

zbioru instancji klasyfikatora specjalizowanego (*przodka*) (rys. 3.2a). W terminologii dotyczącej klas odpowiada to stwierdzeniu, że każdy obiekt podklasy jest jednocześnie obiektem swojej nadklasy (bądź swoich nadklas). W przypadku, gdy klasyfikator specjalizowany jest klasyfikatorem abstrakcyjnym, każda z jego instancji musi być jednocześnie instancją którejś z jego specjalizacji konkretnych (rys. 3.2b).

Związek specjalizacji jest związkiem binarnym. Jeżeli grupa klasyfikatorów ma wspólnego przodka, wówczas z semantyki poszczególnych związków specjalizacji nie wynikają żadne zależności pomiędzy zbiorami instancji klasyfikatorów z tej grupy. Takie własności, jak np. rozłączność zbiorów instancji lub jej brak, mogą być zdefiniowane dopiero dla *zbioru generalizacji* (p. 7.4, str. 82).

3.7 Stan dziedziny

Zbiór instancji klasyfikatora jest zazwyczaj zmienny w czasie. Istniejące w danej chwili instancje nazywa się *ekstensją* klasyfikatora. W modelowaniu pojęciowym, ekstensje klasyfikatorów odpowiadają *stanowi dziedziny*, rozumianemu jako istniejące w danym momencie obiekty i powiązania w dziedzinie problemu.

Definicja 3.7.1 (Stan dziedziny)

Stanem dziedziny (lub krótko *stanem*) jest uporządkowana dwójka:

$$\mathcal{S} = (\text{instances}, \text{ends}),$$

gdzie:

1. $\mathcal{S}.\text{instances}$ jest częściową funkcją ekstensji. Dla danego klasyfikatora funkcja wyznacza zbiór jego instancji:

$$\mathcal{S}.\text{instances}: \text{Classifiers} \rightarrow \mathcal{P}(\text{Instances}). \quad (3.21)$$

2. $\mathcal{S}.\text{ends}$ jest częściową funkcją końców powiązania. Dla danej instancji asocjacji, tj. powiązania, funkcja wyznacza listę — połączonych tym powiązaniem — instancji klas:

$$\mathcal{S}.\text{ends}: \text{Instances} \rightarrow \text{Instances}^{*(2)}. \quad (3.22)$$

Pozycja obiektu na liście identyfikuje koniec powiązania i asocjacji. \square

Definicja 3.7.2 (Zbiór States)

Przez States oznacza się zbiór wszystkich stanów dziedziny postaci jak w definicji 3.7.1. \square

3.8 Relacja spełniania

Pojęciowy diagram klas nakłada na stany dziedziny pewne warunki, bądź z punktu widzenia analityka odzwierciedla warunki, które są własnością dziedziny problemu. Związek pomiędzy pojęciowym diagramem klas i stanem dziedziny definiuje się w postaci relacji *spełniania*: $Sat \subseteq \text{Diagrams} \times \text{States}$. Jeżeli zachodzi $Sat(\mathcal{D}, \mathcal{S})$, to warunki przedstawione na diagramie \mathcal{D} są spełnione w stanie \mathcal{S} .

Zamieszczona dalej formalna definicja relacji Sat opiera się na dwóch założeniach:

1. Diagram nie specyfikuje wszystkich własności dziedziny, tzn. dopuszcza się możliwość istnienia klas i związków, które nie są pokazane na diagramie. Jest to zgodne z ogólną praktyką rysowania diagramów klas¹⁴.
2. W konsekwencji powyższego założenia przyjmuje się, że dwa różne klasyfikatory niepołączone na diagramie związkiem specjalizacji mogą mieć wspólne instancje, chyba że wynika to inaczej z diagramu.

Definicja 3.8.1 (Spełnianie)

Niech $\mathcal{S} \in \text{States}$ i $\mathcal{D} \in \text{Diagrams}$. Diagram \mathcal{D} jest *spełniony w stanie* \mathcal{S} , co zapisuje się:

$$Sat(\mathcal{D}, \mathcal{S}),$$

wtedy i tylko wtedy, gdy:

¹⁴Wspomniane zagadnienie, rozpatrywane w kategoriach *kompletności* i *niekompletności* diagramu, porusza praca [France99].

1. \mathcal{S} specyfikuje ekstensje wszystkich klasyfikatorów \mathcal{D} (oraz być może innych, nie pokazanych na diagramie):

$$\mathcal{D}.\text{classifiers} \subseteq \mathbf{dom}(\mathcal{S}.\text{instances}). \quad (3.23)$$

2. Instancja danej asocjacji łączy tylko instancje klas połączonych tą asocjacją (na określonych jej końcach):

$$\begin{aligned} \forall as \in \mathcal{D}.\text{assocs} \cdot \forall ln \in \mathcal{S}.\text{instances}(as) \cdot & \quad (3.24) \\ \mathbf{len}(\mathcal{D}.\text{ends}(as)) = \mathbf{len}(\mathcal{S}.\text{ends}(ln)) \wedge & \\ \forall i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(as))\} \cdot & \\ \pi_i(\mathcal{S}.\text{ends}(ln)) \in \mathcal{S}.\text{instances}(\pi_i(\mathcal{D}.\text{ends}(as))) & \end{aligned}$$

3. Instancje danej asocjacji spełniają ograniczenia wynikające z liczności jej końców¹⁵. Dla dowolnych $n - 1$ końców asocjacji n -arnej ($n \geq 2$) i $n - 1$ instancji klas na tych końcach, liczba powiązań jakie tworzą one z instancjami klasy na pozostałym końcu należy do liczności tego końca:

$$\forall as \in \mathcal{D}.\text{assocs} \cdot \quad (3.25)$$

$$\begin{aligned} \forall i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(as))\} \cdot \forall p \in \text{product}(as, i) \cdot & \quad (3.26) \\ |\{ln \in \mathcal{S}.\text{instances}(as) : \bar{\pi}_i(\mathcal{S}.\text{ends}(ln)) = p\}| \in & \\ \pi_i(\mathcal{D}.\text{mults}(as)), & \end{aligned}$$

gdzie:

$$\text{product}(as, i) =_{def} \prod_{j=1, j \neq i}^{\mathbf{len}(\mathcal{D}.\text{ends}(as))} \mathcal{S}.\text{instances}(\pi_j(\mathcal{D}.\text{ends}(as))). \quad (3.27)$$

4. Jeżeli asocjacja nie posiada własności $\{\text{bag}\}$, to w skład jej ekstensji wchodzi co najwyżej jedno powiązanie łączące dany zbiór instancji klas (na danych końcach powiązania):

$$\begin{aligned} \forall as \in \mathcal{D}.\text{assocs} \setminus \mathcal{D}.\text{assocs}_{\text{bag}} \cdot \forall ln_1, ln_2 \in \mathcal{S}.\text{instances}(as) \cdot & \quad (3.28) \\ ln_1 \neq ln_2 \Rightarrow \exists i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(as))\} \cdot & \\ \pi_i(\mathcal{S}.\text{ends}(ln_1)) \neq \pi_i(\mathcal{S}.\text{ends}(ln_2)). & \end{aligned}$$

5. Żaden agregat nie jest jednocześnie swoją bezpośrednią lub pośrednią częścią:

$$\forall ob \in \text{Instances} \cdot ob \notin \text{parts}(ob), \quad (3.29)$$

¹⁵Interpretacja liczności końców asocjacji o arności większej niż dwa może być wieloraka, na co wskazuje praca [Genova02]. Interpretacja określona we wspomnianej pracy mianem *potencjalnych krotek* (ang. *potential tuples*) jest niesprzeczna ze specyfikacją UML1, a już bezspornie obowiązującą w specyfikacji UML2 [UML04, str. 37–38]. Taką też prezentuje się tu formalnie.

gdzie parts wyznacza *zbiór części agregatu*. Formalnie:

$$ob_2 \in_{def} \text{parts}(ob_1), \quad (3.30)$$

wtedy i tylko wtedy, gdy ob_2 jest *bezpośrednią* częścią ob_1 :

$$\begin{aligned} \exists as \in \mathcal{D}.\text{assocs}_{agg} \cdot \exists ln \in \mathcal{S}.\text{instances}(as) \cdot \\ ob_1 = \pi_1(\mathcal{S}.\text{ends}(ln)) \wedge ob_2 = \pi_2(\mathcal{S}.\text{ends}(ln)) \end{aligned}$$

lub *pośrednią*, czyli dla pewnego $n \geq 2$ zachodzi:

$$\begin{aligned} \exists as_1, \dots, as_n \in \mathcal{D}.\text{assocs}_{agg} \cdot \\ \exists ln_1 \in \mathcal{S}.\text{instances}(as_1), \dots, \exists ln_n \in \mathcal{S}.\text{instances}(as_n) \cdot \\ ob_1 = \pi_1(\mathcal{S}.\text{ends}(ln_1)) \wedge ob_2 = \pi_2(\mathcal{S}.\text{ends}(ln_n)) \wedge \\ \forall i \in \{1, \dots, n-1\} \cdot \pi_2(\mathcal{S}.\text{ends}(ln_i)) = \pi_1(\mathcal{S}.\text{ends}(ln_{i+1})). \end{aligned}$$

6. Żaden obiekt nie jest jednocześnie bezpośrednią częścią dwóch kompozytów. Precyzyjniej¹⁶, może istnieć tylko jedno powiązanie, będące instancją kompozycji, w którym dany obiekt występuje jako część:

$$\begin{aligned} \forall as_1, as_2 \in \mathcal{D}.\text{assocs}_{com} \cdot \quad (3.31) \\ \forall ln_1 \in \mathcal{S}.\text{instances}(as_1) \cdot \forall ln_2 \in \mathcal{S}.\text{instances}(as_2) \cdot \\ ln_1 \neq ln_2 \Rightarrow \pi_2(\mathcal{S}.\text{ends}(ln_1)) \neq \pi_2(\mathcal{S}.\text{ends}(ln_2)). \end{aligned}$$

7. Jeżeli dany klasyfikator jest specjalizacją innego klasyfikatora, wówczas jego instancja jest również instancją specjalizowanego klasyfikatora:

$$\begin{aligned} \forall cf_1, cf_2 \in \mathcal{D}.\text{classifiers} \cdot cf_2 \in \mathcal{D}.\text{specs}(cf_1) \Rightarrow \quad (3.32) \\ \mathcal{S}.\text{instances}(cf_2) \subseteq \mathcal{S}.\text{instances}(cf_1). \end{aligned}$$

8. Instancja klasyfikatora abstrakcyjnego jest jednocześnie instancją — będącego jego specjalizacją — klasyfikatora konkretnego:

$$\begin{aligned} \forall cf_1 \in \mathcal{D}.\text{abstract} \cdot \forall in \in \mathcal{S}.\text{instances}(cf_1) \cdot \quad (3.33) \\ \exists cf_2 \in \mathcal{D}.\text{classifiers} \setminus \mathcal{D}.\text{abstract} \cdot \\ cf_2 \in \mathcal{D}.\text{specs}(cf_1) \wedge in \in \mathcal{S}.\text{instances}(cf_2). \quad \square \end{aligned}$$

3.9 Znaczenie diagramu

Zbiór stanów dziedziny, które spełniają warunki opisane pojęciowym diagramem klas traktuje się jako interpretację diagramu. Formalnie wyraża to poniższa definicja.

¹⁶Użyte doprecyzowanie oparte jest na pozycji [Rumbaugh04]. Specyfikacja UML2 jest w tej kwestii zbyt ogólna.

Definicja 3.9.1 (Znaczenie diagramu)

Niech $\mathcal{D} \in \text{Diagrams}$. *Znaczeniem* (*interpretacją*) *diagramu* \mathcal{D} jest zbiór:

$$\mathcal{M}(\mathcal{D}),$$

gdzie \mathcal{M} jest funkcją postaci:

$$\begin{aligned} \mathcal{M}: \text{Diagrams} &\rightarrow \mathcal{P}(\text{States}), \\ \mathcal{M}(\mathcal{D}) &=_{def} \{ \mathcal{S} \in \text{States} : \text{Sat}(\mathcal{D}, \mathcal{S}) \}. \quad \square \end{aligned}$$

Zgodnie z przedstawioną wcześniej terminologią dla formalnego języka modelowania (p. 1.4.1, str. 17), funkcja \mathcal{M} w powyższej definicji jest odwzorowaniem semantycznym; zbiór Diagrams — dziedziną syntaktyczną; a zbiór $\mathcal{P}(\text{States})$ — dziedziną semantyczną pojęciowego diagramu klas.

Rozdział 4

Spójność semantyczna

W rozdziale tym przedstawiono zagadnienie spójności semantycznej pojęciowego diagramu klas. Zaproponowane zostało formalne ujęcie tego problemu, zilustrowane przykładem abstrakcyjnych liści w hierarchii specjalizacji oraz agregacji rekursywnej.

4.1 Poprawność a spójność

W ramach oceny jakości dowolnego diagramu w UML rozpatruje się między innymi kryterium jego *poprawności* [Płodzień03]. Wyróżnia się dwa podstawowe rodzaje poprawności:

- *poprawność syntaktyczna* — diagram jest poprawny syntaktycznie, jeżeli przedstawione na nim konstrukcje są zgodne z regułami składniowymi;
- *poprawność semantyczna* — diagram jest poprawny semantycznie, jeżeli odpowiada dziedzinie problemu, którą opisuje.

Reguły poprawności syntaktycznej pojęciowego diagramu klas zawierają się w jego definicji (definicja 3.4.1, str. 26). Poprawność semantyczna pojęciowego diagramu klas nie może być natomiast zdefiniowana i weryfikowana formalnie. Wymagałoby to bowiem istnienia innego (referencyjnego) sformalizowanego opisu dziedziny problemu, który z kolei sam rodziłby pytanie o swą poprawność semantyczną. Poprawność semantyczna może natomiast podlegać weryfikacji nieformalnej, np. w procesie usystematyzowanego przeglądu diagramu [Górski02].

W przeciwieństwie do poprawności, *spójność semantyczna* jest pojęciem dotyczącym diagramu jako takiego i może być rozstrzygnięta bez odwoływania się do zewnętrznych założeń. Rozpatruje się w tym przypadku, czy własności przedstawione na diagramie nie są wzajemnie sprzeczne; innymi słowy, czy może istnieć dziedzina problemu o tych własnościach, a w konsekwencji implementacja diagramu. Naturalnie, implikacją braku spójności semantycznej jest jednoczesny brak poprawności semantycznej.

W oparciu o formalną semantykę pojęciowego diagramu klas jego spójność semantyczna może być zdefiniowana i weryfikowana formalnie. Spójność semantyczną definiuje się dalej najpierw w odniesieniu do pojedynczego klasyfikatora, a następnie dla całego diagramu.

4.2 Spójność klasyfikatora

Klasyfikator jest spójny semantycznie jeżeli własności pokazane na diagramie pozwalają na istnienie instancji tego klasyfikatora. Wyraża to poniższa definicja.

Definicja 4.2.1 (Spójność klasyfikatora)

Niech $\mathcal{D} \in \text{Diagrams}$ i $cf \in \text{Classifiers}$. Klasyfikator cf jest *spójny semantycznie w kontekście diagramu \mathcal{D}* wtedy i tylko wtedy, gdy:

$$\exists \mathcal{S} \in \mathcal{M}(\mathcal{D}) \cdot \mathcal{S}.\text{instances}(cf) \neq \emptyset.$$

W przeciwnym wypadku mówi się, że klasyfikator jest *niespójny semantycznie*. \square

Z definicji spójności klasyfikatora wynikają — ujęte dalej w postaci twierdzeń — proste własności:

- Jeżeli dany klasyfikator jest niespójny semantycznie w kontekście danego diagramu, to niespójne semantycznie są również jego specjalizacje na tym diagramie. I równoważnie¹⁷, jeżeli klasyfikator jest spójny semantycznie, to spójne semantycznie są również klasyfikatory, które specjalizuje.
- Jeżeli dana klasa jest niespójna semantycznie w kontekście danego diagramu, to niespójne semantycznie są również asocjacje, w których bierze udział na tym diagramie. I równoważnie, jeżeli dana asocjacja jest spójna semantycznie, to spójne semantycznie są również klasy na jej końcach.

Twierdzenie 4.2.1 (Spójność specjalizacji)

Niech $\mathcal{D} \in \text{Diagrams}$ i $cf_1, cf_2 \in \mathcal{D}.\text{classifiers}$, takie że:

$$cf_2 \in \mathcal{D}.\text{specs}(cf_1)$$

i klasyfikator cf_1 jest niespójny semantycznie w kontekście diagramu \mathcal{D} . Wówczas klasyfikator cf_2 jest również niespójny semantycznie.

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D})$, wtedy z niespójności semantycznej klasyfikatora cf_1 zachodzi:

$$\mathcal{S}.\text{instances}(cf_1) = \emptyset. \tag{4.1}$$

¹⁷Na mocy *prawa kontrapozycji* [Trzęsicki03].

Ponieważ $Sat(\mathcal{D}, \mathcal{S})$, zatem z punktu 7 definicji Sat (str. 33):

$$\mathcal{S}.instances(cf_2) \subseteq \mathcal{S}.instances(cf_1) \quad (4.2)$$

i ostatecznie:

$$\mathcal{S}.instances(cf_2) = \emptyset. \quad \square \quad (4.3)$$

Twierdzenie 4.2.2 (Spójność asocjacji)

Niech $\mathcal{D} \in \text{Diagrams}$, $cl \in \mathcal{D}.classes$ i $as \in \mathcal{D}.assocs$, takie że dla pewnego $i \in \{1, \dots, \text{len}(\mathcal{D}.ends(as))\}$ zachodzi:

$$\pi_i(\mathcal{D}.ends(as)) = cl$$

i klasa cl jest niespójna semantycznie w kontekście diagramu \mathcal{D} . Wówczas asocjacja as jest również niespójna semantycznie.

Dowód

Niespójność semantyczną asocjacji as dowodzi się nie wprost. Niech istnieje $\mathcal{S} \in \mathcal{M}(\mathcal{D})$ i $ln \in \mathcal{S}.instances(as)$. Z niespójności semantycznej klasy cl zachodzi:

$$\mathcal{S}.instances(cl) = \emptyset. \quad (4.4)$$

Ponieważ $Sat(\mathcal{D}, \mathcal{S})$, zatem z punktu 2 definicji Sat (str. 32):

$$\pi_i(\mathcal{S}.ends(ln)) \in \mathcal{S}.instances(cl), \quad (4.5)$$

co jest sprzeczne z równaniem (4.4). \square

4.3 Abstrakcyjny liść

Najprostszym przykładem klasyfikatora niespójnego semantycznie jest abstrakcyjny klasyfikator nieposiadający na diagramie specjalizacji. O klasyfikatorze bez specjalizacji mówi się, że jest *liściem* w hierarchii specjalizacji.

Definicja 4.3.1 (Liść)

Niech $\mathcal{D} \in \text{Diagrams}$. Klasyfikator $cf \in \mathcal{D}.classifiers$ jest *liściem na diagramie* \mathcal{D} wtedy i tylko wtedy, gdy¹⁸:

$$\mathcal{D}.specs(cf) = \emptyset. \quad \square$$

Twierdzenie 4.3.1 (Niespójność abstrakcyjnego liścia)

Niech $\mathcal{D} \in \text{Diagrams}$ i $cf \in \mathcal{D}.abstract$, takie że klasyfikator cf jest liściem na diagramie \mathcal{D} . Wówczas cf jest niespójny semantycznie w kontekście \mathcal{D} .

¹⁸Zgodnie z tą definicją, do liści zalicza się również klasyfikator niepołączony żadnym związkiem specjalizacji.

Dowód

Niespójność semantyczną klasyfikatora cf dowodzi się nie wprost. Niech istnieje $\mathcal{S} \in \mathcal{M}(\mathcal{D})$ i $in \in \mathcal{S}.instances(cf)$. Ponieważ $Sat(\mathcal{D}, \mathcal{S})$, zatem z punktu 8 definicji Sat (str. 33) istnieje klasyfikator cf_2 , taki że:

$$cf_2 \in \mathcal{D}.specs(cf), \tag{4.6}$$

co jest sprzeczne z założeniem, że cf jest liściem. \square

Jeżeli na pojęciowym diagramie klas występuje abstrakcyjny liść może to oznaczać, że dokonana analiza dziedziny jest błędna — pojęcie powinno być konkretne, albo niekompletna — pojęcie, które zostało zdefiniowane jako abstrakcyjne powinno być doszczegółowione poprzez konkretne specjalizacje.

4.4 Agregacja rekursywna

Agregacja może być *rekursywna* (jak każda asocjacja), czyli taka, w której agregat i część są obiektami tej samej klasy. W przypadku tego typu agregacji, szczególną uwagę należy zwrócić na oznaczenia licznosci końców, gdyż mogą one prowadzić do niespójności semantycznej, co pokazuje poniższy przykład.

Przykład 4.4.1 (Niespójna agregacja rekursywna)

Rysunek 4.1 przedstawia diagram klas zaczerpnięty z pozycji [Miller], zawierający agregację rekursywną. Pokazuje się dalej, że klasa *Wydział* na wspomnianym diagramie jest niespójna semantycznie.

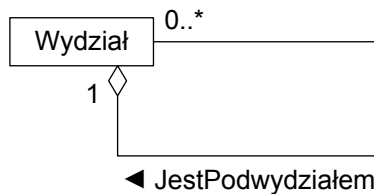
Niech $\mathcal{D} \in \text{Diagrams}$ zawiera konstrukcję pokazaną na rysunku 4.1:

$$JestPodwydziałem \in \mathcal{D}.assocs_{agg}, \tag{4.7}$$

$$\mathcal{D}.ends(JestPodwydziałem) = [Wydział, Wydział], \tag{4.8}$$

$$\mathcal{D}.mults(JestPodwydziałem) = [\{1\}, \mathbb{N}]. \tag{4.9}$$

Niespójność semantyczną klasy *Wydział* dowodzi się nie wprost. Niech istnieje stan $\mathcal{S} \in \mathcal{M}(\mathcal{D})$, dla którego $\mathcal{S}.instances(Wydział) \neq \emptyset$. Niech W oznacza dalej zbiór



Rysunek 4.1: Niespójna agregacja rekursywna

$\mathcal{S}.\text{instances}(\text{Wydział})$, a \sqsubset będzie relacją binarną w zbiorze W ($\sqsubset \subseteq W \times W$) zdefiniowaną następująco:

$$ob_1 \sqsubset ob_2 \Leftrightarrow_{def} ob_1 \in \text{parts}(ob_2), \quad (4.10)$$

gdzie parts jest funkcją, jak w punkcie 5 definicji Sat (str. 32).

Z definicji parts i faktu, że zachodzi $Sat(\mathcal{D}, \mathcal{S})$ łatwo wykazać, że relacja \sqsubset w zbiorze W jest:

- przeciwzwrotna ($\neg ob_1 \sqsubset ob_1$),
- przechodnia ($ob_1 \sqsubset ob_2 \wedge ob_2 \sqsubset ob_3 \Rightarrow ob_1 \sqsubset ob_3$) i
- przeciwsymetryczna ($ob_1 \sqsubset ob_2 \Rightarrow \neg ob_2 \sqsubset ob_1$)¹⁹.

Na mocy powyższych własności, każdy ciąg obiektów ob_1, ob_2, \dots ze zbioru W , spełniający warunek $ob_i \sqsubset ob_{i+1}$ ($i \geq 1$) jest różnoelementowy, tj. $ob_i \neq ob_j$ dla $i \neq j$.

Niech ob_1, \dots, ob_n będzie najdłuższym ciągiem w zbiorze W , takim że $ob_i \sqsubset ob_{i+1}$ ($i \in \{1, \dots, n-1\}$). Skończoność tego ciągu wynika ze skończoności zbioru W i z faktu, że ciąg jest różnoelementowy. Ponieważ nie ma dłuższego ciągu, stąd:

$$\forall ob \in W \cdot \neg ob_n \sqsubset ob. \quad (*)$$

Ponieważ $Sat(\mathcal{D}, \mathcal{S})$, spełniony jest punkt 3 definicji Sat (str. 32). Dla asocjacji $JestPodwydziałem$ i jej końca po stronie agregatu ($i = 1$):

$$\text{product}(JestPodwydziałem, 1) = W \quad (4.11)$$

i dla obiektu ob_n zachodzi:

$$|\{ln \in \mathcal{S}.\text{instances}(JestPodwydziałem) : \bar{\pi}_1(\mathcal{S}.\text{ends}(ln)) = ob_n\}| = 1. \quad (4.12)$$

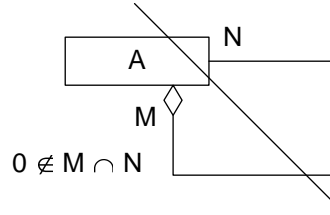
Istnieje zatem powiązanie $ln \in \mathcal{S}.\text{instances}(JestPodwydziałem)$, takie że $\mathcal{S}.\text{ends}(ln) = [ob, ob_n]$, przy czym z punktu 2 definicji Sat (str. 32) $ob \in W$. Ponieważ asocjacja $JestPodwydziałem$ jest agregacją, zatem $ob_n \in \text{parts}(ob)$, co jest sprzeczne z równaniem (*).

Ostatecznie:

$$\forall \mathcal{S} \in \mathcal{M}(\mathcal{D}) \cdot \mathcal{S}.\text{instances}(\text{Wydział}) = \emptyset \quad (4.13)$$

i klasa Wydział jest niespójna semantycznie. \square

¹⁹Jeżeli relacja jest przeciwzwrotna i przechodnia, to jest przeciwsymetryczna [Trzęsicki03].



Rysunek 4.2: Niespójność dla agregacji rekursywnej

Uwaga 4.4.1 (Licznosci końców agregacji rekursywnej)

W przypadku agregacji rekursywnej, takiej jak na rysunku 4.2, licznosci obu jej końców powinny zawierać *zero*. Formalnie:

$$\begin{aligned} \forall as \in \mathcal{D}. \text{assoc}_{\text{agg}} \cdot \pi_1(\mathcal{D}.\text{ends}(as)) = \pi_2(\mathcal{D}.\text{ends}(as)) \Rightarrow \\ 0 \in \pi_1(\mathcal{D}.\text{mults}(as)) \cap \pi_2(\mathcal{D}.\text{mults}(as)). \end{aligned} \tag{4.14}$$

W przeciwnym wypadku, klasa połączona taką agregacją i sama agregacja (twierdzenie 4.2.2) są niespójne semantycznie. Niespójność w przypadku, gdy licznosc końca po stronie części nie zawiera wartości *zero* wykazuje się w analogiczny sposób, jak dla końca po stronie całości w przykładzie 4.4.1. \square

Przykład 4.4.2 (Złożona agregacja rekursywna)

Diagram klas z rysunku 4.1 wymaga, by każdy bez wyjątku wydział był podwydziałem innego wydziału, co jest przyczyną niespójności semantycznej. Właściwym rozwiązaniem, odzwierciedlającym hierarchiczną zależność pomiędzy wydziałami jest diagram z rysunku 4.3. Spójność semantyczną klasyfikatorów na wspomnianym diagramie wykazuje się przez konstrukcję odpowiedniego stanu dziedziny \mathcal{S} , np. takiego że:

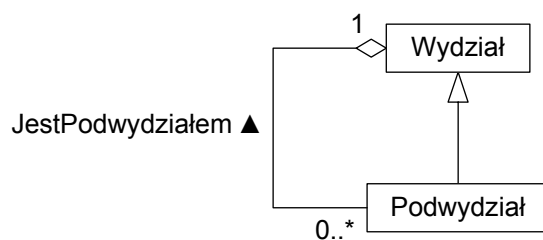
$$\begin{aligned} \mathcal{S}.\text{instances}(\text{Wydział}) &= \{ \text{„Elektroniki i Informatyki”}, \text{„Informatyki”} \}, \\ \mathcal{S}.\text{instances}(\text{Podwydział}) &= \{ \text{„Informatyki”} \}, \\ \mathcal{S}.\text{instances}(\text{JestPodwydziałem}) &= \{ \text{ln} \}, \\ \mathcal{S}.\text{ends}(\text{ln}) &= [\text{„Elektroniki i Informatyki”}, \text{„Informatyki”}]. \end{aligned}$$

W tym przypadku, brak wartości *zero* w licznosci końca po stronie całości nie powoduje niespójności semantycznej, bowiem z semantyki związku specjalizacji nie każdy obiekt klasy *Wydział* musi być obiektem klasy *Podwydział*²⁰. \square

4.5 Spójność diagramu

Diagram jest spójny semantycznie jeżeli własności pokazane na diagramie w żaden sposób nie wykluczają się wzajemnie. Wyraża to poniższa definicja.

²⁰Taka zależność zachodziłaby, gdyby klasa *Wydział* była oznaczona jako abstrakcyjna. Stałaby się wówczas znowu niespójna semantycznie.



Rysunek 4.3: Złożona agregacja rekursywna

Definicja 4.5.1 (Spójność diagramu)

Diagram $\mathcal{D} \in \text{Diagrams}$ jest *spójny semantycznie* wtedy i tylko wtedy, gdy dowolny klasyfikator $cf \in \text{Classifiers}$ jest spójny semantycznie w kontekście tego diagramu. W przeciwnym wypadku mówi się, że diagram jest *niespójny semantycznie*. \square

Jak widać z poprzednich przykładów, reguły poprawności syntaktycznej pojęciowego diagramu klas nie zapewniają jego spójności semantycznej, a tym samym nie gwarantują implementowalności opisanego nim modelu.

Rozdział 5

Wynikanie semantyczne

Rozdział ten definiuje relację wynikania semantycznego w zbiorze wszystkich pojęciowych diagramów klas. Szczegółową analizę przyjętej definicji stanowią przedstawione dalej ogólne reguły wynikania. Rozdział kończy się pokazaniem zależności pomiędzy wynikaniem semantycznym i zaprezentowanym w poprzednim rozdziale problemem spójności semantycznej.

5.1 Relacja wynikania

Matematycznie zdefiniowana semantyka pojęciowego diagramu klas umożliwia formalne wnioskowanie na temat przedstawionych na nim własności dziedziny. Własności, które wynikają z semantyki danego diagramu, a które wyrażone są na nim niejako *implicite*, mogą być pokazane wprost w postaci innego diagramu. Zależność pomiędzy takimi dwoma diagramami definiuje się w postaci relacji *wynikania semantycznego*²¹: $\Rightarrow \subseteq \text{Diagrams} \times \text{Diagrams}$. Jeżeli dla danej dziedziny problemu prawdziwe są własności wyrażone diagramem \mathcal{D}_1 i zachodzi $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, to dla tej dziedziny prawdziwe są również własności przedstawione na diagramie \mathcal{D}_2 . Powyższą relację definiuje się dalej w sposób formalny.

Definicja 5.1.1 (Wynikanie)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$. Diagram \mathcal{D}_2 *wynika semantycznie z diagramu \mathcal{D}_1* , co zapisuje się:

$$\mathcal{D}_1 \Rightarrow \mathcal{D}_2,$$

wtedy i tylko wtedy, gdy:

$$\mathcal{M}(\mathcal{D}_1) \subseteq \mathcal{M}(\mathcal{D}_2),$$

czyli w każdym stanie dziedziny, w którym spełniony jest diagram \mathcal{D}_1 , spełniony jest również diagram \mathcal{D}_2 . \square

²¹Ze względu na analogię tej relacji do implikacji w logice, używa się tu symbolu implikacji (\Rightarrow). Nie prowadzi to jednak dalej do niejednoznaczności.

Twierdzenie 5.1.1 (Własności wynikania)

Relacja wynikania semantycznego \Rightarrow w zbiorze Diagrams jest:

- zwrotna ($\mathcal{D}_1 \Rightarrow \mathcal{D}_1$) i
- przechodnia ($(\mathcal{D}_1 \Rightarrow \mathcal{D}_2 \wedge \mathcal{D}_2 \Rightarrow \mathcal{D}_3) \Rightarrow (\mathcal{D}_1 \Rightarrow \mathcal{D}_3)$).

Dowód

Powyższe własności relacji wynikania są prostą konsekwencją własności relacji zawierania się zbiorów (\subseteq). \square

5.2 Reguły wynikania

Przechodniość relacji wynikania semantycznego pozwala wnioskować o pojęciowym diagramie klas w procesie złożonym z wielu łatwiejszych do wykazania kroków. Podstawą takiego wielokrokowego wnioskowania może być zatem pewien zbiór stosunkowo prostych wzorców przekształceń diagramów — reguł wynikania semantycznego.

Dalej definiuje się w postaci twierdzeń łącznie dziesięć reguł wynikania, pogrupowanych w podrozdziały ze względu na element diagramu, którego dotyczą.

5.3 Reguły wynikania dla klasy

W grupie reguł dotyczących klasy definiuje się regułę:

- usunięcie klasy.

Twierdzenie 5.3.1 (Usunięcie klasy)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &\neq \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.1) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

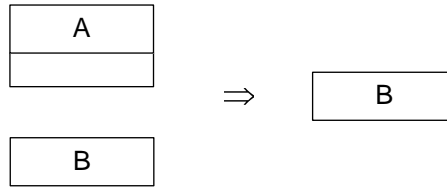
i diagramy różnią się obecnością jednej klasy:

$$A \in \mathcal{D}_1.\text{classes} \setminus \mathcal{D}_2.\text{classes}, \quad (5.2)$$

$$\mathcal{D}_2.\text{classes} = \mathcal{D}_1.\text{classes} \setminus \{A\}. \quad (5.3)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.1)²².

²²Z założeń twierdzenia wynika, że klasa A nie jest na diagramie \mathcal{D}_1 połączona żadną asocjacją (w tym nie ma atrybutów), bądź związkiem specjalizacji oraz nie jest klasą abstrakcyjną.



Rysunek 5.1: Usunięcie klasy

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 1 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.1).

Ponieważ $Sat(\mathcal{D}_1, \mathcal{S})$, zatem zachodzi:

$$\mathcal{D}_1.\text{classifiers} \subseteq \mathbf{dom}(\mathcal{S}.\text{instances}), \quad (5.4)$$

a z założeń twierdzenia:

$$\mathcal{D}_2.\text{classifiers} \subset \mathcal{D}_1.\text{classifiers} \quad (5.5)$$

i ostatecznie:

$$\mathcal{D}_2.\text{classifiers} \subset \mathbf{dom}(\mathcal{S}.\text{instances}). \quad \square \quad (5.6)$$

5.4 Reguły wynikania dla asocjacji

W grupie reguł dotyczących asocjacji definiuje się reguły:

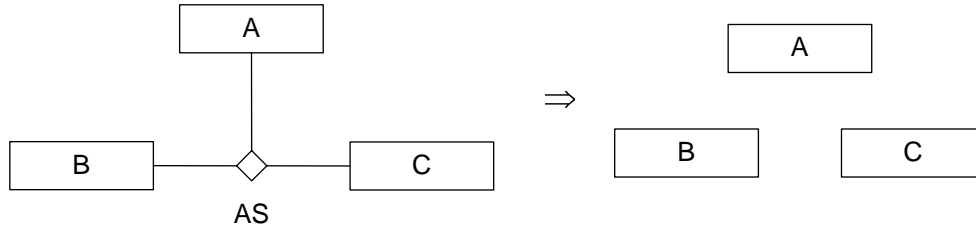
- usunięcie asocjacji,
- rozszerzenie liczności końca asocjacji,
- usunięcie unikalności powiązań.

Za wyjątkiem pierwszej z nich, reguły te obejmują również przypadek klasy asocjacji, agregacji lub kompozycji.

Twierdzenie 5.4.1 (Usunięcie asocjacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.7) \\ \mathcal{D}_1.\text{assocs} &\neq \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\ \mathcal{D}_1.\text{ends} &\neq \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\ \mathcal{D}_1.\text{mults} &\neq \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\ \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}}, \end{aligned}$$



Rysunek 5.2: Usunięcie asocjacji

i diagramy różnią się obecnością jednej asocjacji:

$$AS \in \mathcal{D}_1.\text{assocs} \setminus \mathcal{D}_1.\text{classes}, \quad (5.8)$$

$$\mathcal{D}_2.\text{assocs} = \mathcal{D}_1.\text{assocs} \setminus \{AS\}. \quad (5.9)$$

Dla pozostałych asocjacji, klasy na ich końcach oraz licznosci końców nie różnią się na obu diagramach:

$$\forall as \in \mathcal{D}_2.\text{assocs} \cdot \mathcal{D}_2.\text{ends}(as) = \mathcal{D}_1.\text{ends}(as), \quad (5.10)$$

$$\forall as \in \mathcal{D}_2.\text{assocs} \cdot \mathcal{D}_2.\text{mults}(as) = \mathcal{D}_1.\text{mults}(as). \quad (5.11)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.2)²³.

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 1 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założeń twierdzenia.

Na podstawie $Sat(\mathcal{D}_1, \mathcal{S})$ oraz założeń twierdzenia wykazuje się, analogicznie jak w dowodzie twierdzenia 5.3.1, że zachodzi:

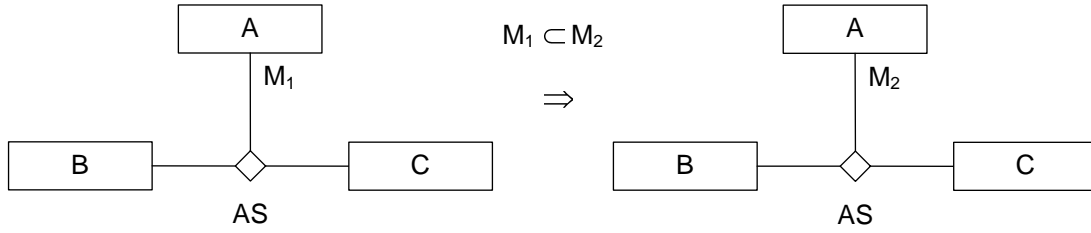
$$\mathcal{D}_2.\text{classifiers} \subset \mathbf{dom}(\mathcal{S}.\text{instances}). \quad \square \quad (5.12)$$

Twierdzenie 5.4.2 (Rozszerzenie licznosci końca asocjacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.13) \\ \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\ \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\ \mathcal{D}_1.\text{mults} &\neq \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\ \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}}, \end{aligned}$$

²³Z założeń twierdzenia wynika, że asocjacja AS nie jest na diagramie \mathcal{D}_1 agregacją ani kompozycją, nie jest połączona związkiem specjalizacji oraz nie jest abstrakcyjna.



Rysunek 5.3: Rozszerzenie licznosci końca asocjacji

i diagramy zawierają asocjację $AS \in \mathcal{D}_1.\text{assocs}$, taką że licznosc jej końca k na diagramie \mathcal{D}_1 jest właściwym podzbiorem licznosci tego końca na diagramie \mathcal{D}_2 :

$$\pi_k(\mathcal{D}_1.\text{mults}(AS)) \subset \pi_k(\mathcal{D}_2.\text{mults}(AS)), \quad (5.14)$$

a licznosci pozostałych końców asocjacji AS oraz końców innych asocjacji nie różnią się na obu diagramach:

$$\bar{\pi}_k(\mathcal{D}_1.\text{mults}(AS)) = \bar{\pi}_k(\mathcal{D}_2.\text{mults}(AS)), \quad (5.15)$$

$$\forall as \in \mathcal{D}_1.\text{assocs} \setminus \{AS\} \cdot \mathcal{D}_1.\text{mults}(as) = \mathcal{D}_2.\text{mults}(as). \quad (5.16)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.3).

Dowód

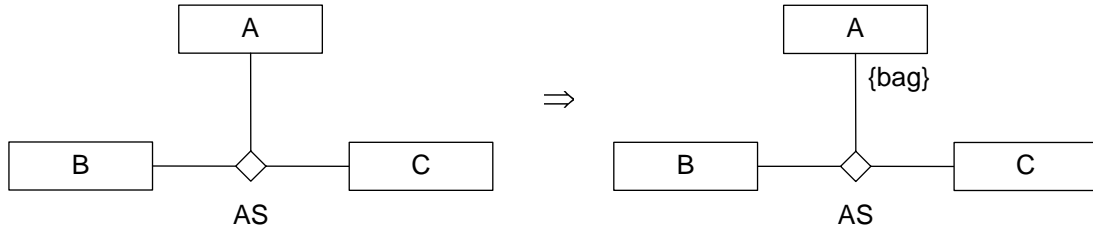
Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 3 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.13).

Niech $as \in \mathcal{D}_2.\text{assocs}$, $i \in \{1, \dots, \text{len}(\mathcal{D}_2.\text{ends}(as))\}$ i $p \in \text{product}(as, i)$ (z założenia (5.13) i definicji funkcji product , ma ona jednakową postać dla diagramów \mathcal{D}_1 i \mathcal{D}_2). Jeżeli $as \neq AS$ lub $i \neq k$, to wspomniany punkt spełniony jest z $Sat(\mathcal{D}_1, \mathcal{S})$ oraz równań (5.15) i (5.16). W przeciwnym wypadku, z $Sat(\mathcal{D}_1, \mathcal{S})$ zachodzi:

$$\begin{aligned} |\{ln \in \mathcal{S}.\text{instances}(AS) : \bar{\pi}_k(\mathcal{S}.\text{ends}(ln)) = p\}| \in \\ \pi_k(\mathcal{D}_1.\text{mults}(AS)) \end{aligned} \quad (5.17)$$

i z założenia (5.14):

$$\begin{aligned} |\{ln \in \mathcal{S}.\text{instances}(AS) : \bar{\pi}_k(\mathcal{S}.\text{ends}(ln)) = p\}| \in \\ \pi_k(\mathcal{D}_2.\text{mults}(AS)). \quad \square \end{aligned} \quad (5.18)$$



Rysunek 5.4: Usunięcie unikalności powiązań

Twierdzenie 5.4.3 (Usunięcie unikalności powiązań)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocS}_{\text{agg}} &= \mathcal{D}_2.\text{assocS}_{\text{agg}}, & (5.19) \\ \mathcal{D}_1.\text{assocS} &= \mathcal{D}_2.\text{assocS}, & \mathcal{D}_1.\text{assocS}_{\text{com}} &= \mathcal{D}_2.\text{assocS}_{\text{com}}, \\ \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\ \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\ \mathcal{D}_1.\text{assocS}_{\text{bag}} &\neq \mathcal{D}_2.\text{assocS}_{\text{bag}}, \end{aligned}$$

i diagramy zawierają asocjację $AS \in \mathcal{D}_1.\text{assocS}$, która na diagramie \mathcal{D}_1 nie posiada własności $\{\text{bag}\}$, a na diagramie \mathcal{D}_2 posiada taką własność. Pozostałe asocjacje nie różnią się na obu diagramach:

$$AS \in \mathcal{D}_2.\text{assocS}_{\text{bag}}, \tag{5.20}$$

$$\mathcal{D}_1.\text{assocS}_{\text{bag}} = \mathcal{D}_2.\text{assocS}_{\text{bag}} \setminus \{AS\}. \tag{5.21}$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.4).

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 4 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.19).

Niech $as \in \mathcal{D}_2.\text{assocS} \setminus \mathcal{D}_2.\text{assocS}_{\text{bag}}$. Z założeń (5.20) i (5.21):

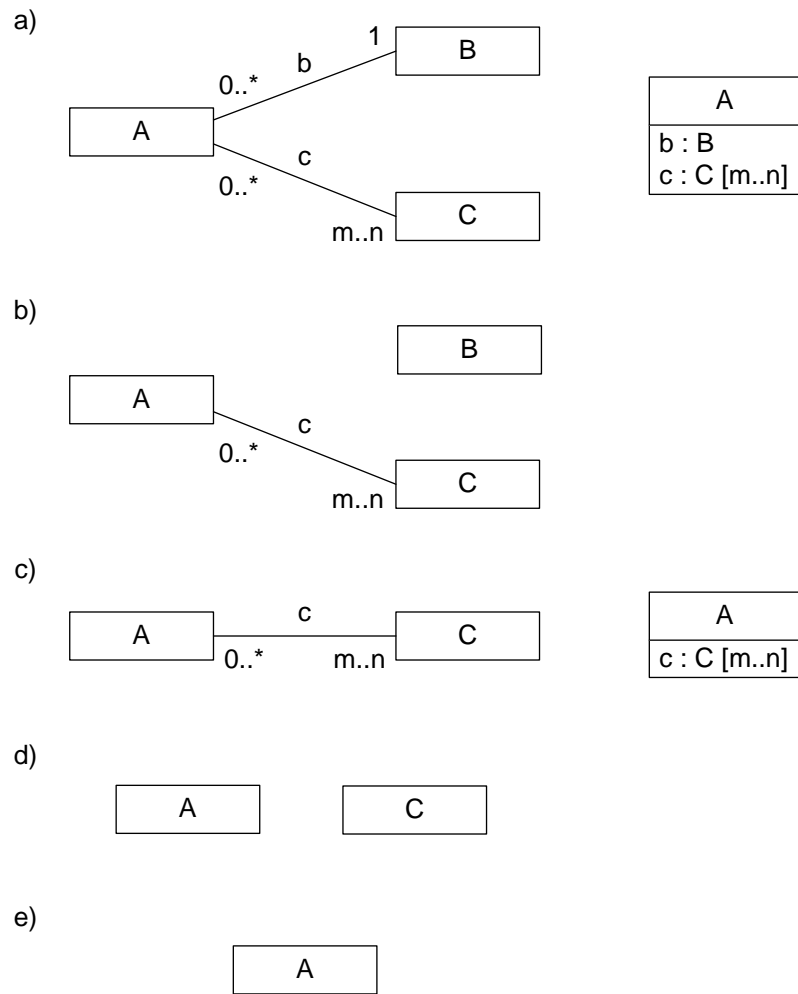
$$\mathcal{D}_1.\text{assocS}_{\text{bag}} \subset \mathcal{D}_2.\text{assocS}_{\text{bag}} \tag{5.22}$$

i $as \in \mathcal{D}_1.\text{assocS} \setminus \mathcal{D}_1.\text{assocS}_{\text{bag}}$. Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, zatem wspomniany punkt jest spełniony. \square

Przykład 5.4.1 (Usunięcie atrybutów)

Rysunek 5.5 przedstawia następujący ciąg związków wynikania semantycznego:

1. a) \Rightarrow b) — usunięcie asocjacji b (twierdzenie 5.4.1),
2. b) \Rightarrow c) — usunięcie klasy B (twierdzenie 5.3.1),



Rysunek 5.5: Usunięcie atrybutów

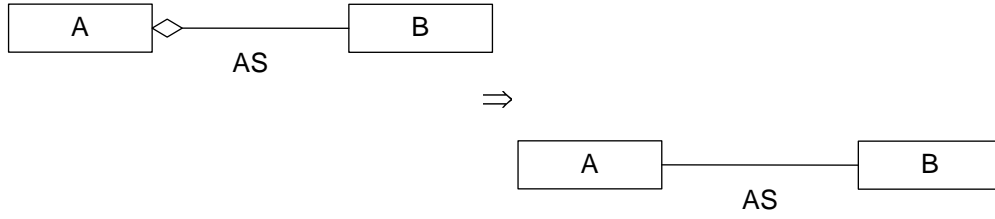
3. c) \Rightarrow d) — usunięcie asocjacji c (twierdzenie 5.4.1),
4. d) \Rightarrow e) — usunięcie klasy C (twierdzenie 5.3.1). \square

5.5 Reguły wynikania dla agregacji i kompozycji

W grupie reguł dotyczących agregacji i kompozycji definiuje się reguły:

- osłabienie agregacji i
- osłabienie kompozycji.

Reguły te obejmują również przypadek klasy asocjacji.



Rysunek 5.6: Osłabienie agregacji

Twierdzenie 5.5.1 (Osłabienie agregacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &\neq \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.23) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają asocjację $AS \in \mathcal{D}_1.\text{assocs}$, która jest agregacją na diagramie \mathcal{D}_1 , a nie jest nią na diagramie \mathcal{D}_2 . Pozostałe asocjacje nie różnią się na obu diagramach:

$$AS \in \mathcal{D}_1.\text{assocs}_{\text{agg}}, \quad (5.24)$$

$$\mathcal{D}_2.\text{assocs}_{\text{agg}} = \mathcal{D}_1.\text{assocs}_{\text{agg}} \setminus \{AS\}. \quad (5.25)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.6).

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $\text{Sat}(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 5 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $\text{Sat}(\mathcal{D}_1, \mathcal{S})$ i założenia (5.23).

Niech parts_1 i parts_2 wyznaczają części agregatu w stanie \mathcal{S} , odpowiednio dla diagramu \mathcal{D}_1 i \mathcal{D}_2 . Dalej dowodzi się nie wprost. Niech istnieje $ob \in \text{Instances}$, takie że:

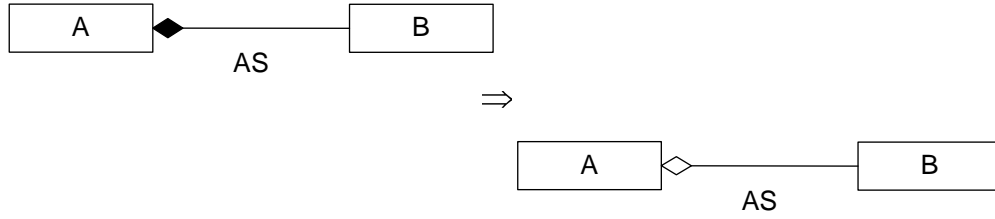
$$ob \in \text{parts}_2(ob). \quad (5.26)$$

Z założeń (5.24) i (5.25):

$$\mathcal{D}_2.\text{assocs}_{\text{agg}} \subset \mathcal{D}_1.\text{assocs}_{\text{agg}} \quad (5.27)$$

i z definicji funkcji parts_1 oraz parts_2 zachodzi:

$$\forall ob \in \text{Instances} \cdot \text{parts}_2(ob) \subseteq \text{parts}_1(ob). \quad (5.28)$$



Rysunek 5.7: Osłabienie kompozycji

A zatem:

$$ob \in \text{parts}_1(ob), \quad (5.29)$$

ale z faktu, że zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$ wynika:

$$ob \notin \text{parts}_1(ob). \quad (5.30)$$

Otrzymana sprzeczność oznacza niepoprawność założenia (5.26). \square

Twierdzenie 5.5.2 (Osłabienie kompozycji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.31) \\ \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &\neq \mathcal{D}_2.\text{assocs}_{\text{com}}, \\ \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\ \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\ \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}}, \end{aligned}$$

i diagramy zawierają agregację $AS \in \mathcal{D}_1.\text{assocs}_{\text{agg}}$, która na diagramie \mathcal{D}_1 jest kompozycją, a nie jest nią na diagramie \mathcal{D}_2 . Pozostałe agregacje nie różnią się na obu diagramach:

$$AS \in \mathcal{D}_1.\text{assocs}_{\text{com}}, \quad (5.32)$$

$$\mathcal{D}_2.\text{assocs}_{\text{com}} = \mathcal{D}_1.\text{assocs}_{\text{com}} \setminus \{AS\}. \quad (5.33)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.7).

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 6 definicji Sat (str. 33). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.31).

Niech $as_1, as_2 \in \mathcal{D}_2.\text{assocs}_{\text{com}}$. Z założeń (5.32) i (5.33):

$$\mathcal{D}_2.\text{assocs}_{\text{com}} \subset \mathcal{D}_1.\text{assocs}_{\text{com}} \quad (5.34)$$

i $as_1, as_2 \in \mathcal{D}_1.\text{assocs}_{\text{com}}$. Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, zatem wspomniany punkt jest spełniony. \square

5.6 Reguły wynikania dla klasy asocjacji

W grupie reguł dotyczących klasy asocjacji definiuje się regułę:

- przekształcenie klasy asocjacji w klasę²⁴.

Twierdzenie 5.6.1 (Przekształcenie klasy asocjacji w klasę)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.35) \\
 \mathcal{D}_1.\text{assocs} &\neq \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &\neq \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &\neq \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają klasę $D \in \mathcal{D}_1.\text{classes}$, która jest klasą asocjacji na diagramie \mathcal{D}_1 , a nie jest nią na diagramie \mathcal{D}_2 . Pozostałe klasy nie różnią się na obu diagramach:

$$D \in \mathcal{D}_1.\text{assocs}, \quad (5.36)$$

$$\mathcal{D}_2.\text{assocs} = \mathcal{D}_1.\text{assocs} \setminus \{D\}. \quad (5.37)$$

Dla pozostałych asocjacji, klasy na ich końcach oraz liczności końców również nie różnią się na obu diagramach:

$$\forall as \in \mathcal{D}_2.\text{assocs} \cdot \mathcal{D}_2.\text{ends}(as) = \mathcal{D}_1.\text{ends}(as), \quad (5.38)$$

$$\forall as \in \mathcal{D}_2.\text{assocs} \cdot \mathcal{D}_2.\text{mults}(as) = \mathcal{D}_1.\text{mults}(as). \quad (5.39)$$

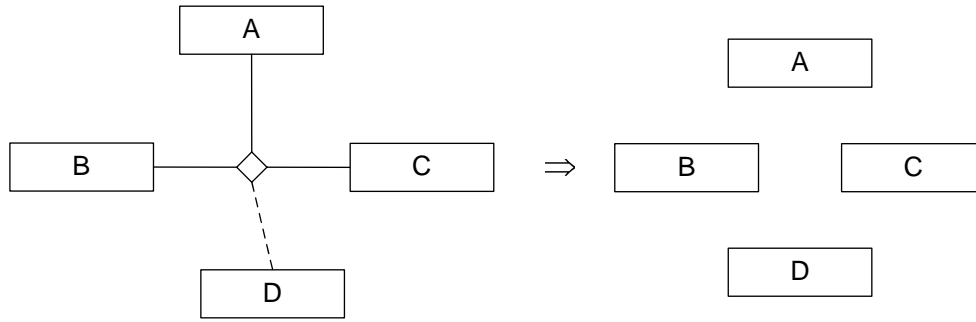
Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.8)²⁵.

Dowód

Powyższe twierdzenie dowodzi się analogicznie, jak twierdzenie 5.4.1 (str. 44). W tym przypadku zachodzi równość zbiorów klasyfikatorów na obu diagramach, tj. $\mathcal{D}_1.\text{classifiers} = \mathcal{D}_2.\text{classifiers}$. \square

²⁴Semantyczną zależność pomiędzy szczególnym przypadkiem klasy asocjacji i asocjacją pokazuje rozdział na temat równoważności semantycznej (p. 6.10, str. 77).

²⁵Z założeń twierdzenia wynika, że na diagramie \mathcal{D}_1 , klasa asocjacji D nie jest agregacją ani kompozycją oraz może specjalizować jedynie klasę — por. punkt 8 definicji pojęciowego diagramu klas (str. 28).



Rysunek 5.8: Przekształcenie klasy asocjacji w klasę

5.7 Reguły wynikania dla związku specjalizacji

W grupie reguł dotyczących związku specjalizacji definiuje się reguły:

- usunięcie abstrakcyjności klasyfikatora,
- usunięcie związku specjalizacji i
- promocja asocjacji.

Ostatnia reguła obejmuje również przypadek klasy asocjacji, agregacji lub kompozycji.

Twierdzenie 5.7.1 (Usunięcie abstrakcyjności klasyfikatora)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (5.40) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &\neq \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają klasyfikator $A \in \mathcal{D}_1.\text{classifiers}$, który na diagramie \mathcal{D}_1 jest klasyfikatorem abstrakcyjnym, a na diagramie \mathcal{D}_2 konkretnym. Pozostałe klasyfikatory nie różnią się na obu diagramach:

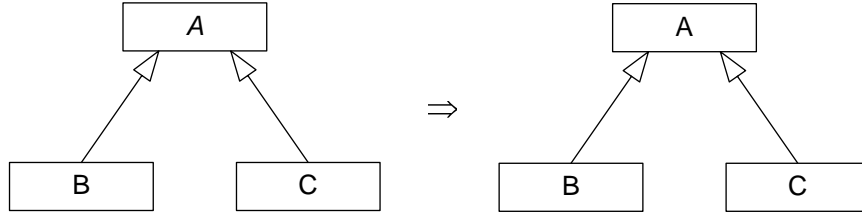
$$A \in \mathcal{D}_1.\text{abstract}, \quad (5.41)$$

$$\mathcal{D}_2.\text{abstract} = \mathcal{D}_1.\text{abstract} \setminus \{A\}. \quad (5.42)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.9 — dla klasy).

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $\text{Sat}(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie,



Rysunek 5.9: Usunięcie abstrakcyjności klasyfikatora

że spełniony jest punkt 8 definicji Sat (str. 33). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.40).

Niech $cf_1 \in \mathcal{D}_2.\text{abstract}$ i $in \in \mathcal{S}.\text{instances}(cf_1)$. Z założeń (5.41) i (5.42):

$$\mathcal{D}_2.\text{abstract} \subset \mathcal{D}_1.\text{abstract}, \quad (5.43)$$

a zatem $cf_1 \in \mathcal{D}_1.\text{abstract}$. Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, istnieje klasyfikator $cf_2 \in \mathcal{D}_2.\text{classifiers} \setminus \mathcal{D}_1.\text{abstract}$, taki że:

$$cf_2 \in \mathcal{D}_2.\text{specs}(cf_1) \wedge in \in \mathcal{S}.\text{instances}(cf_2). \quad (5.44)$$

Z równania (5.43):

$$\mathcal{D}_2.\text{classifiers} \setminus \mathcal{D}_1.\text{abstract} \subset \mathcal{D}_2.\text{classifiers} \setminus \mathcal{D}_2.\text{abstract}, \quad (5.45)$$

stąd $cf_2 \in \mathcal{D}_2.\text{classifiers} \setminus \mathcal{D}_2.\text{abstract}$. \square

Twierdzenie 5.7.2 (Usunięcie związku specjalizacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocS}_{\text{agg}} &= \mathcal{D}_2.\text{assocS}_{\text{agg}}, & (5.46) \\ \mathcal{D}_1.\text{assocS} &= \mathcal{D}_2.\text{assocS}, & \mathcal{D}_1.\text{assocS}_{\text{com}} &= \mathcal{D}_2.\text{assocS}_{\text{com}}, \\ \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &\neq \mathcal{D}_2.\text{specs}, \\ \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\ \mathcal{D}_1.\text{assocS}_{\text{bag}} &= \mathcal{D}_2.\text{assocS}_{\text{bag}}, \end{aligned}$$

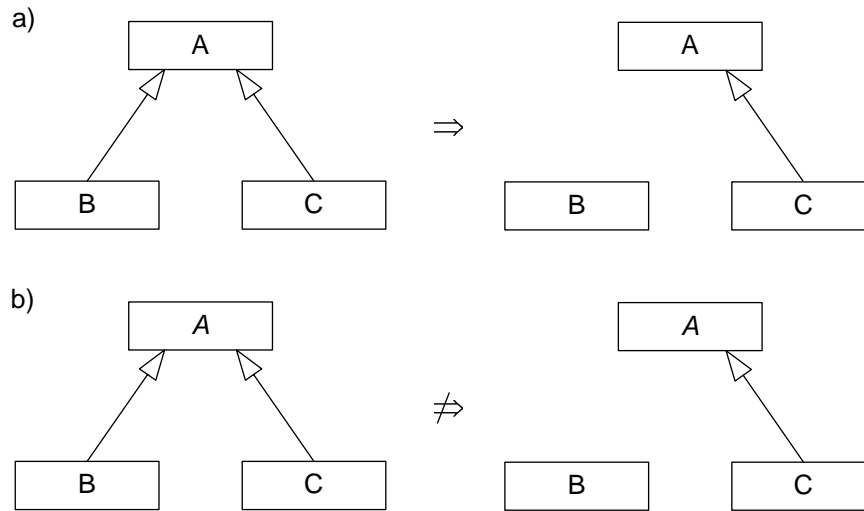
i diagramy zawierają klasyfikatory $A, B \in \mathcal{D}_1.\text{classifiers}$, takie że B jest specjalizacją A na diagramie \mathcal{D}_1 :

$$B \in \mathcal{D}_1.\text{specs}(A), \quad (5.47)$$

natomiast na diagramie \mathcal{D}_2 nie łączy je związek specjalizacji. Pozostałe związki specjalizacji nie różnią się na obu diagramach:

$$\mathcal{D}_2.\text{specs}(A) = \mathcal{D}_1.\text{specs}(A) \setminus \{B\}, \quad (5.48)$$

$$\forall cf \in \mathcal{D}_1.\text{classifiers} \setminus \{A\} \cdot \mathcal{D}_2.\text{specs}(cf) = \mathcal{D}_1.\text{specs}(cf). \quad (5.49)$$



Rysunek 5.10: Usunięcie związku specjalizacji

Niech ponadto A będzie klasyfikatorem konkretnym:

$$A \notin \mathcal{D}_1.\text{abstract}. \quad (5.50)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.10a).

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełnione są punkty 7 i 8 definicji Sat (str. 33). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.46).

Punkt 7

Niech $cf_1, cf_2 \in \mathcal{D}_2.\text{classifiers}$ i $cf_2 \in \mathcal{D}_2.\text{specs}(cf_1)$. Z założeń (5.48) i (5.49) wynika:

$$cf_2 \in \mathcal{D}_1.\text{specs}(cf_1). \quad (5.51)$$

Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, zatem powyższy punkt jest spełniony.

Punkt 8

Niech $cf_1 \in \mathcal{D}_2.\text{abstract}$ i $in \in \mathcal{S}.\text{instances}(cf_1)$. Ponieważ $Sat(\mathcal{D}_1, \mathcal{S})$, zatem istnieje klasyfikator $cf_2 \in \mathcal{D}_2.\text{classifiers} \setminus \mathcal{D}_2.\text{abstract}$, taki że:

$$cf_2 \in \mathcal{D}_1.\text{specs}(cf_1) \wedge in \in \mathcal{S}.\text{instances}(cf_2). \quad (5.52)$$

Z założenia (5.50):

$$cf_1 \neq A, \quad (5.53)$$

wobec czego z założenia (5.49) wynika:

$$\mathcal{D}_1.\text{specs}(cf_1) = \mathcal{D}_2.\text{specs}(cf_1), \quad (5.54)$$

a stąd $cf_2 \in \mathcal{D}_2.\text{specs}(cf_1)$. \square

Uwaga 5.7.1 (Usunięcie związku specjalizacji a abstrakcyjność)

W przypadku, gdy klasyfikator specjalizowany jest klasyfikatorem abstrakcyjnym, powyższe wynikanie nie zachodzi (rys. 5.10b). Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$ będą diagramami, jak w powyższym twierdzeniu, przy czym niech A będzie klasyfikatorem abstrakcyjnym:

$$A \in \mathcal{D}_1.\text{abstract}. \quad (5.55)$$

Rozpatruje się dalej stan $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$, taki że:

$$\begin{aligned} \mathcal{S}.\text{instances}(A) &= \{b, c\}, \\ \mathcal{S}.\text{instances}(B) &= \{b\}, \\ \mathcal{S}.\text{instances}(C) &= \{c\}. \end{aligned} \quad (5.56)$$

W tym stanie, dla obiektu b w kontekście diagramu \mathcal{D}_2 zachodzi:

$$\begin{aligned} \neg \exists cf_2 \in \mathcal{D}_2.\text{classifiers} \setminus \mathcal{D}_2.\text{abstract}. \\ cf_2 \in \mathcal{D}_2.\text{specs}(A) \wedge b \in \mathcal{S}.\text{instances}(cf_2). \end{aligned} \quad (5.57)$$

Zatem $\mathcal{S} \notin \mathcal{M}(\mathcal{D}_2)$, gdyż nie jest spełniony punkt 8 definicji *Sat* (str. 33). W konsekwencji diagram \mathcal{D}_2 nie może wynikać semantycznie z diagramu \mathcal{D}_1 . \square

Twierdzenie 5.7.3 (Promocja asocjacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, \\ \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\ \mathcal{D}_1.\text{ends} &\neq \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\ \mathcal{D}_1.\text{mults} &\neq \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\ \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}}, \end{aligned} \quad (5.58)$$

i diagramy zawierają klasy $A, B \in \mathcal{D}_1.\text{classes}$, takie że B jest specjalizacją A :

$$B \in \mathcal{D}_1.\text{specs}(A), \quad (5.59)$$

oraz asocjację $AS \in \mathcal{D}_1.\text{assocs}$, której koniec k łączy na diagramie \mathcal{D}_1 klasę B :

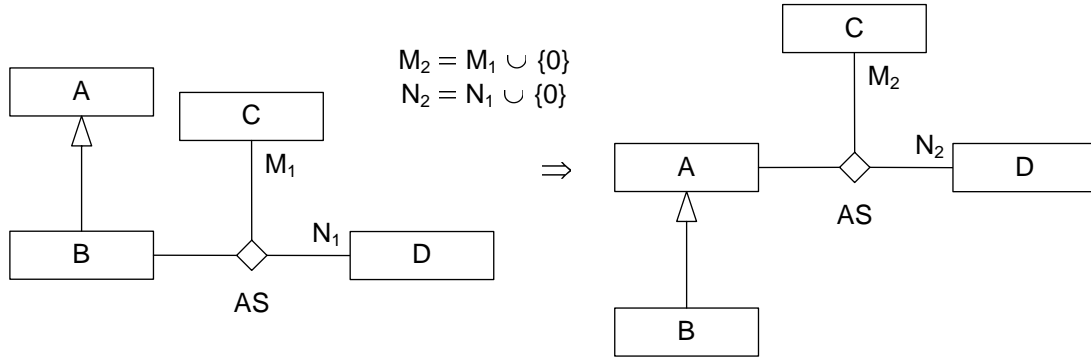
$$\pi_k(\mathcal{D}_1.\text{ends}(AS)) = B, \quad (5.60)$$

a na diagramie \mathcal{D}_2 klasę A :

$$\pi_k(\mathcal{D}_2.\text{ends}(AS)) = A. \quad (5.61)$$

Klasy na pozostałych końcach asocjacji AS nie różnią się na obu diagramach:

$$\bar{\pi}_k(\mathcal{D}_1.\text{ends}(AS)) = \bar{\pi}_k(\mathcal{D}_2.\text{ends}(AS)). \quad (5.62)$$



Rysunek 5.11: Promocja asocjacji

Liczność końca k asocjacji AS nie różni się na obu diagramach:

$$\pi_k(\mathcal{D}_1.\text{mults}(AS)) = \pi_k(\mathcal{D}_2.\text{mults}(AS)), \quad (5.63)$$

a w przypadku pozostałych końców rozszerzona jest na diagramie \mathcal{D}_2 o wartość 0:

$$\begin{aligned} \forall i \in \{1, \dots, \mathbf{len}(\mathcal{D}_2.\text{ends}(AS))\} \setminus \{k\}. \\ \pi_i(\mathcal{D}_2.\text{mults}(AS)) = \pi_i(\mathcal{D}_1.\text{mults}(AS)) \cup \{0\}. \end{aligned} \quad (5.64)$$

Dla pozostałych asocjacji, klasy na ich końcach oraz licznosci końców nie różnią się na obu diagramach:

$$\forall as \in \mathcal{D}_1.\text{assocs} \setminus \{AS\} \cdot \mathcal{D}_1.\text{ends}(as) = \mathcal{D}_2.\text{ends}(as), \quad (5.65)$$

$$\forall as \in \mathcal{D}_1.\text{assocs} \setminus \{AS\} \cdot \mathcal{D}_1.\text{mults}(as) = \mathcal{D}_2.\text{mults}(as). \quad (5.66)$$

Wówczas diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 (rys. 5.11).

Dowód

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełnione są punkty 2 i 3 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założeń twierdzenia.

Punkt 2

Niech $as \in \mathcal{D}_2.\text{assocs}$ i $ln \in \mathcal{S}.\text{instances}(as)$. Jeżeli $as \neq AS$, to powyższy punkt spełniony jest z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.65). Dalej rozpatruje się przypadek, gdy $as = AS$. Z założenia (5.62):

$$\mathbf{len}(\mathcal{D}_1.\text{ends}(AS)) = \mathbf{len}(\mathcal{D}_2.\text{ends}(AS)) \quad (5.67)$$

i ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, to:

$$\mathbf{len}(\mathcal{D}_2.\text{ends}(AS)) = \mathbf{len}(\mathcal{S}.\text{ends}(ln)). \quad (5.68)$$

Do wykazania pozostaje warunek:

$$\begin{aligned} \forall i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\mathbf{ends}(AS))\}. \\ \pi_i(\mathcal{S}.\mathbf{ends}(ln)) \in \mathcal{S}.\mathbf{instances}(\pi_i(\mathcal{D}.\mathbf{ends}(AS))). \end{aligned} \quad (5.69)$$

Niech $i \in \{1, \dots, \mathbf{len}(\mathcal{D}_2.\mathbf{ends}(AS))\}$. Jeżeli $i \neq k$, to powyższy warunek spełniony jest z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (5.62). Pozostaje zatem przypadek, gdy $i = k$.

Ponieważ $Sat(\mathcal{D}_1, \mathcal{S})$, stąd:

$$\pi_k(\mathcal{S}.\mathbf{ends}(ln)) \in \mathcal{S}.\mathbf{instances}(B), \quad (5.70)$$

a z założenia (5.59) i z punktu 7 definicji Sat (str. 33):

$$\mathcal{S}.\mathbf{instances}(B) \subseteq \mathcal{S}.\mathbf{instances}(A). \quad (5.71)$$

Ostatecznie:

$$\pi_k(\mathcal{S}.\mathbf{ends}(ln)) \in \mathcal{S}.\mathbf{instances}(A). \quad (5.72)$$

Punkt 3

Niech $as \in \mathcal{D}_2.\mathbf{assocs}$. Jeżeli $as \neq AS$, to powyższy punkt spełniony jest z $Sat(\mathcal{D}_1, \mathcal{S})$ oraz założeń (5.65) i (5.66). Dalej rozpatruje się przypadek, gdy $as = AS$.

Niech $i \in \{1, \dots, \mathbf{len}(\mathcal{D}_2.\mathbf{ends}(AS))\}$ i $p \in \mathbf{product}_2(AS, i)$, gdzie:

$$\mathbf{product}_2(as, i) = \prod_{j=1, j \neq i}^{\mathbf{len}(\mathcal{D}_2.\mathbf{ends}(as))} \mathcal{S}.\mathbf{instances}(\pi_j(\mathcal{D}_2.\mathbf{ends}(as))). \quad (5.73)$$

Przyjmując dalej oznaczenie:

$$\alpha(i, p) =_{def} |\{ ln \in \mathcal{S}.\mathbf{instances}(AS) : \bar{\pi}_i(\mathcal{S}.\mathbf{ends}(ln)) = p \}|, \quad (5.74)$$

należy wykazać własność:

$$\alpha(i, p) \in \pi_i(\mathcal{D}_2.\mathbf{mults}(AS)). \quad (*)$$

Z założenia (5.62):

$$\mathbf{len}(\mathcal{D}_1.\mathbf{ends}(AS)) = \mathbf{len}(\mathcal{D}_2.\mathbf{ends}(AS)). \quad (5.75)$$

Ponieważ $Sat(\mathcal{D}_1, \mathcal{S})$, zachodzi:

$$\forall p \in \mathbf{product}_1(AS, i) \cdot \alpha(i, p) \in \pi_i(\mathcal{D}_1.\mathbf{mults}(AS)), \quad (5.76)$$

gdzie:

$$\mathbf{product}_1(as, i) = \prod_{j=1, j \neq i}^{\mathbf{len}(\mathcal{D}_1.\mathbf{ends}(as))} \mathcal{S}.\mathbf{instances}(\pi_j(\mathcal{D}_1.\mathbf{ends}(as))). \quad (5.77)$$

Dalej rozpatruje się następujące przypadki:

1. Jeżeli $i = k$, wówczas z założenia (5.62):

$$\text{product}_1(AS, i) = \text{product}_2(AS, i) \quad (5.78)$$

i z równania (5.76) zachodzi:

$$\alpha(i, p) \in \pi_i(\mathcal{D}_1.\text{mults}(AS)). \quad (5.79)$$

Ostatecznie, z założenia (5.63) wynika własność (*).

2. Jeżeli $i \neq k$

(a) oraz $p \in \text{product}_1(AS, i)$, wówczas dowodzi się dalej z równania (5.76), jak dla przypadku $i = k$.

(b) oraz $p \notin \text{product}_1(AS, i)$, wówczas dla l zdefiniowanego jak niżej:

$$l = \begin{cases} k, & \text{jeżeli } i > k \\ k - 1, & \text{jeżeli } i < k \end{cases} \quad (5.80)$$

zachodzi:

$$\pi_l(p) \notin \mathcal{S}.\text{instances}(\pi_k(\mathcal{D}_1.\text{ends}(AS))). \quad (5.81)$$

Prawdziwa jest ponadto, łatwa do wykazania, równość:

$$\pi_l(\bar{\pi}_i(\mathcal{S}.\text{ends}(ln))) = \pi_k(\mathcal{S}.\text{ends}(ln)). \quad (5.82)$$

Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, z punktu 2 definicji Sat (str. 32):

$$\begin{aligned} \forall ln \in \mathcal{S}.\text{instances}(AS) \cdot \\ \pi_l(\bar{\pi}_i(\mathcal{S}.\text{ends}(ln))) \in \mathcal{S}.\text{instances}(\pi_k(\mathcal{D}_1.\text{ends}(AS))), \end{aligned} \quad (5.83)$$

wobec czego, na podstawie równania (5.81):

$$\forall ln \in \mathcal{S}.\text{instances}(AS) \cdot \bar{\pi}_i(\mathcal{S}.\text{ends}(ln)) \neq p \quad (5.84)$$

i w konsekwencji:

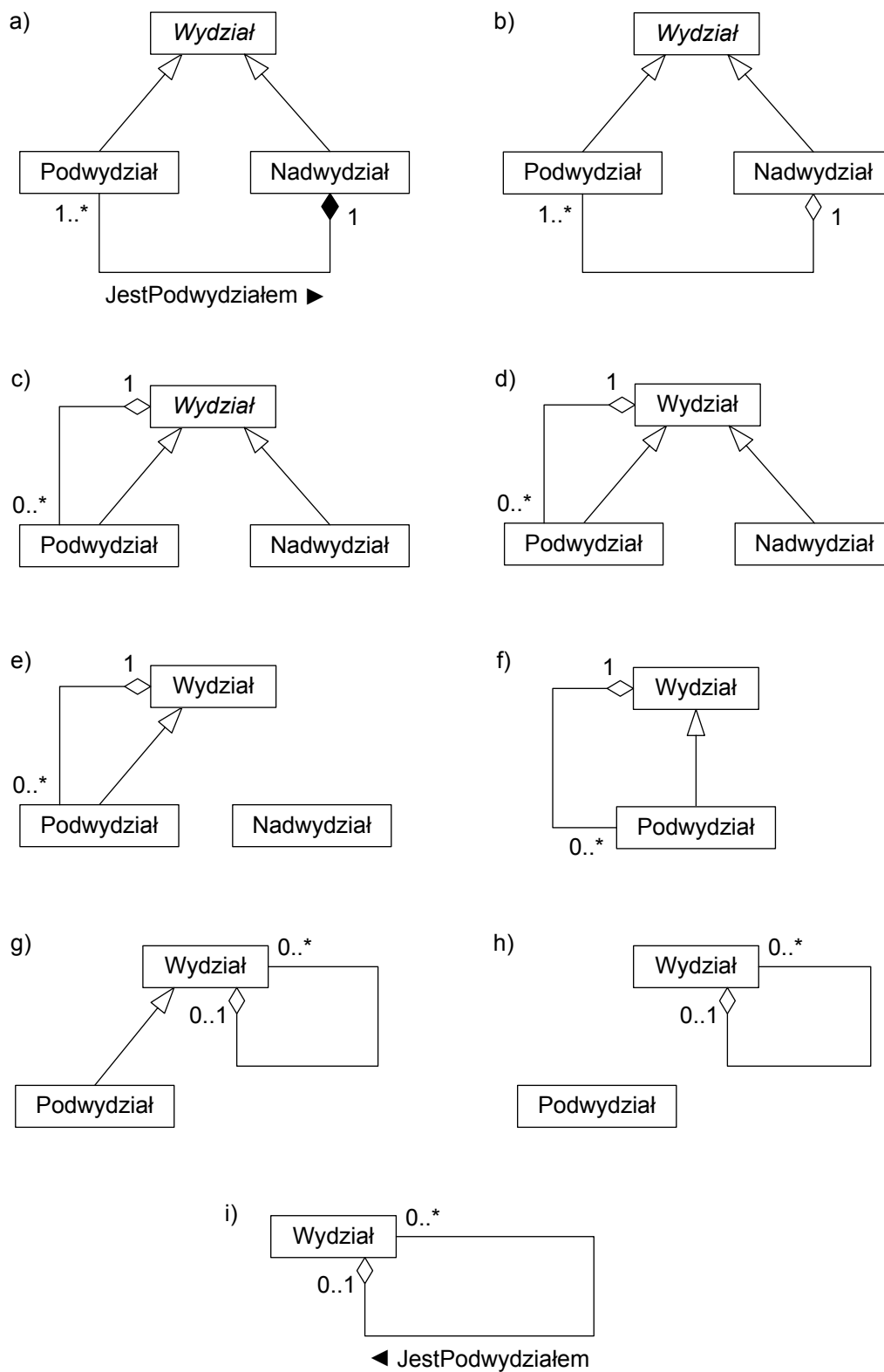
$$\alpha(i, p) = 0. \quad (5.85)$$

Ostatecznie, z założenia (5.64) wynika własność (*). \square

Przykład 5.7.1 (Promocja agregacji)

Rysunek 5.12 przedstawia następujący ciąg związków wynikania semantycznego:

1. a) \Rightarrow b) — osłabienie kompozycji *JestPodwydziałem* (twierdzenie 5.5.2),
2. b) \Rightarrow c) — promocja agregacji *JestPodwydziałem* (twierdzenie 5.7.3),



Rysunek 5.12: Promocja agregacji

3. $c) \Rightarrow d)$ — usunięcie abstrakcyjności klasy *Wydział* (twierdzenie 5.7.1),
4. $d) \Rightarrow e)$ — usunięcie związku specjalizacji (twierdzenie 5.7.2),
5. $e) \Rightarrow f)$ — usunięcie klasy *Nadwydział* (twierdzenie 5.3.1),
6. $f) \Rightarrow g)$ — promocja agregacji *JestPodwydziałem* (twierdzenie 5.7.3),
7. $g) \Rightarrow h)$ — usunięcie związku specjalizacji (twierdzenie 5.7.2),
8. $h) \Rightarrow i)$ — usunięcie klasy *Podwydział* (twierdzenie 5.3.1). \square

Uwaga 5.7.2 (Wynikanie a uściślanie)

Powyższy przykład obrazuje pewną praktyczną własność relacji wynikania semantycznego. Każdy z diagramów na rysunku 5.12(a–i) może stanowić opis tej samej dziedziny problemu, opracowany z różną szczegółowością (np. odpowiadającą postępowi procesu konstrukcji modelu pojęciowego). Jeżeli zachodzi $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, to diagram \mathcal{D}_1 jest nie mniej szczegółowym (kompletnym, precyzyjnym) niż diagram \mathcal{D}_2 opisem danej dziedziny. Innymi słowy, \mathcal{D}_1 może być traktowany jako *uściślenie* (ang. *refinement*) \mathcal{D}_2 [Szlenk03]. \square

5.8 Wynikanie a spójność semantyczna

Intuicyjnie, ze spójnego semantycznie opisu dziedziny nie powinno wynikać nic, co nie jest spójne semantycznie. Spójność semantyczna danego klasyfikatora lub diagramu charakteryzuje również płynące z nich wnioski, co formalnie potwierdzają poniższe twierdzenia.

Twierdzenie 5.8.1 (Spójność wniosku)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$ i $cf \in \text{Classifiers}$, takie że:

$$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$$

i klasyfikator cf jest spójny semantycznie w kontekście diagramu \mathcal{D}_1 . Wówczas cf jest również spójny semantycznie w kontekście diagramu \mathcal{D}_2 .

Dowód

Ze spójności semantycznej klasyfikatora cf w kontekście diagramu \mathcal{D}_1 , istnieje stan dziedziny $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$, taki że:

$$\mathcal{S}.\text{instances}(cf) \neq 0. \tag{5.86}$$

Ponieważ diagram \mathcal{D}_2 wynika semantycznie z diagramu \mathcal{D}_1 , zatem:

$$\mathcal{M}(\mathcal{D}_1) \subseteq \mathcal{M}(\mathcal{D}_2), \tag{5.87}$$

a stąd $\mathcal{S} \in \mathcal{M}(\mathcal{D}_2)$. \square

Twierdzenie 5.8.2 (Spójność wniosku*)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$$

i diagram \mathcal{D}_1 jest spójny semantycznie. Wówczas diagram \mathcal{D}_2 jest również spójny semantycznie.

Dowód

Powyższe twierdzenie wynika z definicji spójności semantycznej diagramu oraz twierdzenia 5.8.1. \square

5.9 Złożona agregacja rekursywna

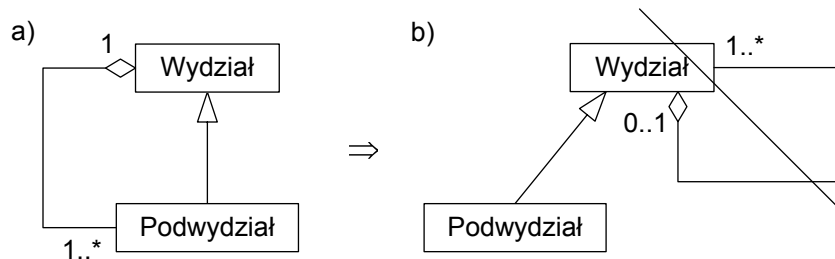
Punkt 4.4 (str. 38) wskazywał na ryzyko zaistnienia niespójności semantycznej w przypadku agregacji rekursywnej. Jednocześnie proponuje się tam spójne semantycznie rozwiązanie wykorzystujące złożoną agregację rekursywną, w której agregat i część nie są tą samą klasą (ale mają wspólne obiekty). W przypadku takiej agregacji, ryzyko niespójności semantycznej jest jednak wciąż aktualne, co pokazuje poniższy przykład.

Przykład 5.9.1 (Niespójna złożona agregacja rekursywna)

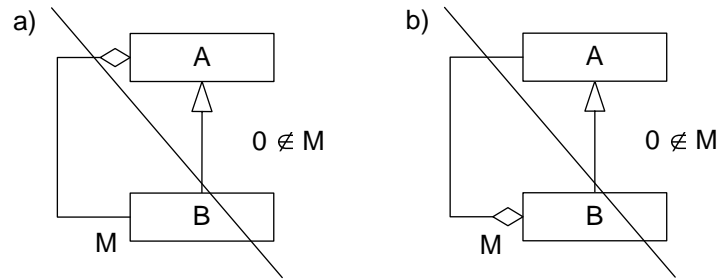
Rysunek 5.13a przedstawia diagram, który na pierwszy rzut oka nie budzi podejrzeń. Jednak promocja agregacji (twierdzenie 5.7.3) prowadzi do diagramu 5.13b, a zgodnie z uwagą 4.4.1 (str. 40) oraz twierdzeniem 4.2.1 (str. 36) klasa *Wydział*, łącząca ją agregacja oraz klasa *Podwydział* są w jego kontekście niespójne semantycznie. Zatem na mocy twierdzenia 5.8.1 (lub 5.8.2) powyższe klasyfikatory są niespójne semantycznie również w kontekście diagramu 5.13a. \square

Uwaga 5.9.1 (Liczności końców złożonej agregacji rekursywnej)

W przypadku złożonej agregacji rekursywnej, takiej jak na rysunku 5.14a



Rysunek 5.13: Niespójność we wniosku semantycznym



Rysunek 5.14: Niespójność dla złożonej agregacji rekursywnej

lub 5.14b, licznosc końca po stronie specjalizacji powinna zawierać *zero*. Formalnie:

$$\begin{aligned} \forall as \in \mathcal{D}.assoc_{agg} \cdot \forall cl_1, cl_2 \in \mathcal{D}.classes \cdot & \quad (5.88) \\ (\mathcal{D}.ends(as) = [cl_1, cl_2] \wedge cl_2 \in \mathcal{D}.specs(cl_1)) \Rightarrow & \\ 0 \in \pi_2(\mathcal{D}.mults(as)) & \end{aligned}$$

oraz

$$\begin{aligned} \forall as \in \mathcal{D}.assoc_{agg} \cdot \forall cl_1, cl_2 \in \mathcal{D}.classes \cdot & \quad (5.89) \\ (\mathcal{D}.ends(as) = [cl_2, cl_1] \wedge cl_2 \in \mathcal{D}.specs(cl_1)) \Rightarrow & \\ 0 \in \pi_1(\mathcal{D}.mults(as)). & \end{aligned}$$

W przeciwnym wypadku, klasy połączone taką agregacją oraz sama agregacja są niespójne semantycznie. Niespójność w obu przypadkach wykazuje się tak, jak w przykładzie 5.9.1. \square

Rozdział 6

Równoważność semantyczna

W niniejszym rozdziale definiuje się relację równoważności semantycznej w zbiorze pojęciowych diagramów klas. Odwołując się do tej relacji wskazuje się między innymi na możliwość konstruowania w UML nieprecyzyjnych modeli pojęciowych, w których częściowo ukryte pozostają pewne istotne własności dziedziny problemu.

6.1 Relacja równoważności

Szczególnym przypadkiem związku wynikania semantycznego są diagramy o tym samym znaczeniu. Takie przypadki opisuje dalej relacja *równoważności semantycznej*: $\Leftrightarrow \subseteq \text{Diagrams} \times \text{Diagrams}$. Jeżeli zachodzi $\mathcal{D}_1 \Leftrightarrow \mathcal{D}_2$, to diagramy \mathcal{D}_1 i \mathcal{D}_2 są całkowicie zamiennymi opisami pewnej dziedziny problemu (lub jej fragmentu).

Definicja 6.1.1 (Równoważność)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$. Diagram \mathcal{D}_1 jest *równoważny semantycznie diagramowi* \mathcal{D}_2 , co zapisuje się:

$$\mathcal{D}_1 \Leftrightarrow \mathcal{D}_2,$$

wtedy i tylko wtedy, gdy:

$$\mathcal{M}(\mathcal{D}_1) = \mathcal{M}(\mathcal{D}_2),$$

czyli diagram \mathcal{D}_1 jest spełniony w danym stanie wtedy i tylko wtedy, gdy spełniony jest w nim również diagram \mathcal{D}_2 . \square

Twierdzenie 6.1.1 (Własności równoważności)

Relacja równoważności semantycznej \Leftrightarrow w zbiorze Diagrams jest:

- zwrotna ($\mathcal{D}_1 \Leftrightarrow \mathcal{D}_1$),
- symetryczna ($(\mathcal{D}_1 \Leftrightarrow \mathcal{D}_2) \Rightarrow (\mathcal{D}_2 \Leftrightarrow \mathcal{D}_1)$) i

- przechodnia $((\mathcal{D}_1 \Leftrightarrow \mathcal{D}_2 \wedge \mathcal{D}_2 \Leftrightarrow \mathcal{D}_3) \Rightarrow (\mathcal{D}_1 \Leftrightarrow \mathcal{D}_3))^{26}$.

Dowód

Powyższe własności relacji równoważności semantycznej są prostą konsekwencją własności relacji równości zbiorów (=). \square

Twierdzenie 6.1.2 (Alternatywna definicja równoważności)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$. Diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 wtedy i tylko wtedy, gdy \mathcal{D}_2 wynika semantycznym z \mathcal{D}_1 i \mathcal{D}_1 wynika semantycznym z \mathcal{D}_2 , czyli:

$$\mathcal{D}_1 \Leftrightarrow \mathcal{D}_2 \text{ wtedy i tylko wtedy, gdy } \mathcal{D}_1 \Rightarrow \mathcal{D}_2 \text{ i } \mathcal{D}_2 \Rightarrow \mathcal{D}_1.$$

Dowód

Z własności równości zbiorów zachodzi:

$$\mathcal{M}(\mathcal{D}_1) = \mathcal{M}(\mathcal{D}_2) \Leftrightarrow \mathcal{M}(\mathcal{D}_1) \subseteq \mathcal{M}(\mathcal{D}_2) \wedge \mathcal{M}(\mathcal{D}_2) \subseteq \mathcal{M}(\mathcal{D}_1). \quad \square \quad (6.1)$$

6.2 Precyzyjność diagramu

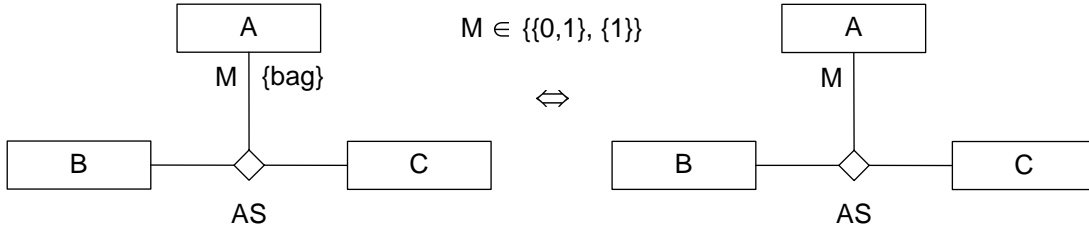
Warunki, które reprezentują poszczególne elementy na pojęciowym diagramie klas, dotyczą obiektów i/lub powiązań w dziedzinie problemu i mogą być od siebie w pewien sposób zależne. Znaczenie diagramu jest złożeniem tych warunków, dlatego też ostateczna interpretacja elementów i oznaczeń może być odmienna od przypadku, w którym interpretuje się je jako odrębne elementy składni. To z kolei może prowadzić do sytuacji, w których pewne oznaczenia nie mają w efekcie żadnego wpływu na semantykę diagramu, bądź są zbyt mało precyzyjne w stosunku do jego semantyki. Oba przypadki są potencjalnym źródłem nieporozumień.

Powyższy problem w pewnym zakresie rozwiązuje uzupełnianie metamodelu UML o kolejne ograniczenia, niepozwalające na tworzenie „nieprecyzyjnych” konstrukcji. Kilka obszarów ewentualnego rozszerzenia metamodelu UML prezentuje się w dalszej części rozdziału.

6.3 Unikalność powiązań a licznosc

Jeżeli którykolwiek z końców asocjacji oznaczony jest licznoscia, do której należy wartość większa niż *jeden*, to z semantyki licznosci, powiazania, wchodzące w skład ekstensji takiej asocjacji, będą unikalne ze względu na łączone przez nie obiekty. Oznaczenie asocjacji własnością {bag} nie wpłynie zatem na semantykę diagramu.

²⁶Jest zatem relacją równoważności w sensie teoriomnogościowym [Trzęsicki03].



Rysunek 6.1: Unikalność powiązań a licznosc

Twierdzenie 6.3.1 (Unikalność powiązań a licznosc)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocS}_{\text{agg}} &= \mathcal{D}_2.\text{assocS}_{\text{agg}}, & (6.2) \\
 \mathcal{D}_1.\text{assocS} &= \mathcal{D}_2.\text{assocS}, & \mathcal{D}_1.\text{assocS}_{\text{com}} &= \mathcal{D}_2.\text{assocS}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocS}_{\text{bag}} &\neq \mathcal{D}_2.\text{assocS}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają asocjację $AS \in \mathcal{D}_1.\text{assocS}$, taką że licznosc jej końca k spełnia warunek:

$$\pi_k(\mathcal{D}_1.\text{mults}(AS)) \in \{\{0, 1\}, \{1\}\}, \quad (6.3)$$

przy czym na diagramie \mathcal{D}_1 AS posiada własność $\{bag\}$, a na diagramie \mathcal{D}_2 nie posiada takiej własności. Pozostałe asocjacje nie różnią się na obu diagramach:

$$AS \in \mathcal{D}_1.\text{assocS}_{\text{bag}}, \quad (6.4)$$

$$\mathcal{D}_2.\text{assocS}_{\text{bag}} = \mathcal{D}_1.\text{assocS}_{\text{bag}} \setminus \{AS\}. \quad (6.5)$$

Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.1).

Dowód

Dowodzi się najpierw, że zachodzi wynikanie $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, a następnie $\mathcal{D}_2 \Rightarrow \mathcal{D}_1$.

$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $\text{Sat}(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 4 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $\text{Sat}(\mathcal{D}_1, \mathcal{S})$ i założenia (6.2).

Niech $as \in \mathcal{D}_2.\text{assocS} \setminus \mathcal{D}_2.\text{assocS}_{\text{bag}}$. Jeżeli $as \neq AS$, to $as \in \mathcal{D}_1.\text{assocS} \setminus \mathcal{D}_1.\text{assocS}_{\text{bag}}$ i wspomniany punkt spełniony jest z $\text{Sat}(\mathcal{D}_1, \mathcal{S})$. Pozostaje zatem wykazać, że:

$$\begin{aligned}
 \forall ln_1, ln_2 \in \mathcal{S}.\text{instances}(AS). & \quad (*) \\
 ln_1 \neq ln_2 &\Rightarrow \exists i \in \{1, \dots, \text{len}(\mathcal{D}.\text{ends}(AS))\}. \\
 \pi_i(\mathcal{S}.\text{ends}(ln_1)) &\neq \pi_i(\mathcal{S}.\text{ends}(ln_2)).
 \end{aligned}$$

Dalej dowodzi się nie wprost. Niech istnieją powiązania $ln_1, ln_2 \in \mathcal{S}.instances(AS)$, takie że:

$$ln_1 \neq ln_2 \wedge \mathcal{S}.ends(ln_1) = \mathcal{S}.ends(ln_2). \quad (6.6)$$

Wówczas dla $p = \bar{\pi}_k(\mathcal{S}.ends(ln_1))$:

$$|\{ ln \in \mathcal{S}.instances(AS) : \bar{\pi}_i(\mathcal{S}.ends(ln)) = p \}| \geq 2, \quad (6.7)$$

co oznacza sprzeczność z $Sat(\mathcal{D}_1, \mathcal{S})$, gdyż z punktu 3 definicji Sat (str. 32) i założenia (6.3) zachodzi:

$$|\{ ln \in \mathcal{S}.instances(AS) : \bar{\pi}_i(\mathcal{S}.ends(ln)) = p \}| \in \{0, 1\}. \quad (6.8)$$

Otrzymana sprzeczność oznacza prawdziwość własności (*).

$\mathcal{D}_2 \Rightarrow \mathcal{D}_1$

Spełnione są założenia twierdzenia 5.4.3 (str. 46). \square

6.4 Kompozycja a licznosc

W przypadku związku kompozycji, w dowolnym stanie dziedziny dany obiekt może być częścią tylko jednego kompozytu²⁷. W związku z tym, ewentualne wartości większe niż *jeden*, należące do licznosci końca kompozycji po stronie kompozytu, są nieosiągalne wśród stanów dziedziny i mogą być na diagramie zignorowane.

Twierdzenie 6.4.1 (Kompozycja a licznosc)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.classes &= \mathcal{D}_2.classes, & \mathcal{D}_1.assoc_{agg} &= \mathcal{D}_2.assoc_{agg}, & (6.9) \\ \mathcal{D}_1.assoc &= \mathcal{D}_2.assoc, & \mathcal{D}_1.assoc_{com} &= \mathcal{D}_2.assoc_{com}, \\ \mathcal{D}_1.ends &= \mathcal{D}_2.ends, & \mathcal{D}_1.specs &= \mathcal{D}_2.specs, \\ \mathcal{D}_1.mults &\neq \mathcal{D}_2.mults, & \mathcal{D}_1.abstract &= \mathcal{D}_2.abstract \\ \mathcal{D}_1.assoc_{bag} &= \mathcal{D}_2.assoc_{bag}, \end{aligned}$$

i diagramy zawierają kompozycję $AS \in \mathcal{D}_1.assoc_{com}$, taką że pomiędzy licznosciami jej pierwszego końca na obu diagramach zachodzi zależność:

$$\pi_1(\mathcal{D}_2.mults(AS)) = \pi_1(\mathcal{D}_1.mults(AS)) \cap \{0, 1\}, \quad (6.10)$$

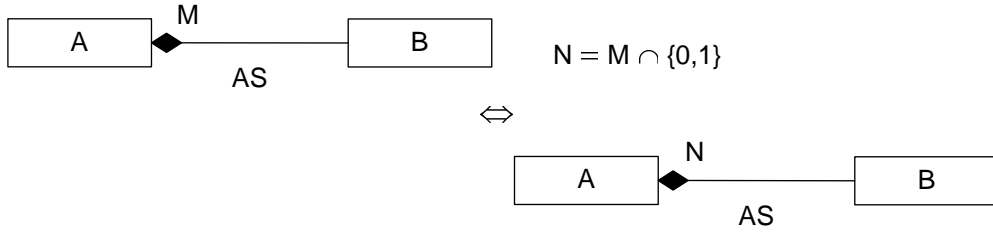
a licznosci drugiego końca kompozycji AS oraz końców innych asocjacji nie różnią się na obu diagramach:

$$\pi_2(\mathcal{D}_1.mults(AS)) = \pi_2(\mathcal{D}_2.mults(AS)), \quad (6.11)$$

$$\forall as \in \mathcal{D}_1.assoc \setminus \{AS\} \cdot \mathcal{D}_1.mults(as) = \mathcal{D}_2.mults(as). \quad (6.12)$$

Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.2).

²⁷Przy czym nie musi być to ciągle ten sam kompozyt, a dany obiekt może również przestać być częścią jakiegokolwiek kompozytu [UML04, str. 38].



Rysunek 6.2: Kompozycja a licznosc

Dowód

Dowodzi się najpierw, że zachodzi $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, a następnie $\mathcal{D}_2 \Rightarrow \mathcal{D}_1$.

$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 3 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (6.9).

Niech $as \in \mathcal{D}_2.\text{assoc}$, $i \in \{1, \dots, \text{len}(\mathcal{D}_2.\text{ends}(as))\}$ i $p \in \text{product}(as, i)$ (z założenia (6.9) i definicji funkcji product , ma ona jednakową postać dla diagramów \mathcal{D}_1 i \mathcal{D}_2). Jeżeli $as \neq AS$ lub $i \neq 1$, to wspomniany punkt spełniony jest z $Sat(\mathcal{D}_1, \mathcal{S})$ oraz równań (6.11) i (6.12). Przyjmując dalej oznaczenie:

$$\alpha(i, p) =_{\text{def}} |\{ ln \in \mathcal{S}.\text{instances}(AS) : \bar{\pi}_i(\mathcal{S}.\text{ends}(ln)) = p \}|, \quad (6.13)$$

pozostaje wykazać własność:

$$\alpha(1, p) \in \pi_1(\mathcal{D}_2.\text{mults}(AS)), \quad (*)$$

gdzie:

$$p \in \mathcal{S}.\text{instances}(\pi_2(\mathcal{D}.\text{ends}(AS))). \quad (6.14)$$

Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, zatem:

$$\alpha(1, p) \in \pi_1(\mathcal{D}_1.\text{mults}(AS)), \quad (6.15)$$

a z punktu 6 definicji Sat (str. 33):

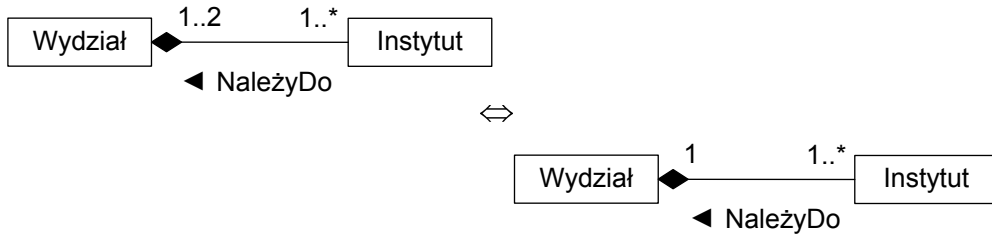
$$\begin{aligned} \forall ln_1, ln_2 \in \mathcal{S}.\text{instances}(AS) \cdot \\ ln_1 \neq ln_2 \Rightarrow \pi_2(\mathcal{S}.\text{ends}(ln_1)) \neq \pi_2(\mathcal{S}.\text{ends}(ln_2)). \end{aligned} \quad (6.16)$$

Dalej dowodzi się nie wprost. Niech istnieje $m \geq 2$, takie że:

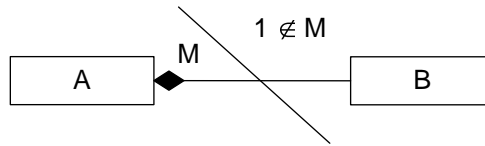
$$\alpha(1, p) = m. \quad (6.17)$$

Istnieją zatem powiązania $ln_1, ln_2 \in \mathcal{S}.\text{instances}(AS)$, takie że:

$$ln_1 \neq ln_2 \wedge \pi_2(\mathcal{S}.\text{ends}(ln_1)) = \pi_2(\mathcal{S}.\text{ends}(ln_2)) = p, \quad (6.18)$$



Rysunek 6.3: Nieprecyzyjna liczność końca kompozycji



Rysunek 6.4: Niespójność dla kompozycji

co prowadzi do sprzeczności z równaniem (6.16). Stąd:

$$\alpha(1, p) \in \{0, 1\}, \quad (6.19)$$

a z równania (6.15):

$$\alpha(1, p) \in \pi_1(\mathcal{D}_1.\text{mults}(AS)) \cap \{0, 1\}, \quad (6.20)$$

i własność (*) jest spełniona przy założeniu (6.10).

$\mathcal{D}_2 \Rightarrow \mathcal{D}_1$

Z założeń twierdzenia:

$$\pi_1(\mathcal{D}_2.\text{mults}(AS)) \subseteq \pi_1(\mathcal{D}_1.\text{mults}(AS)). \quad (6.21)$$

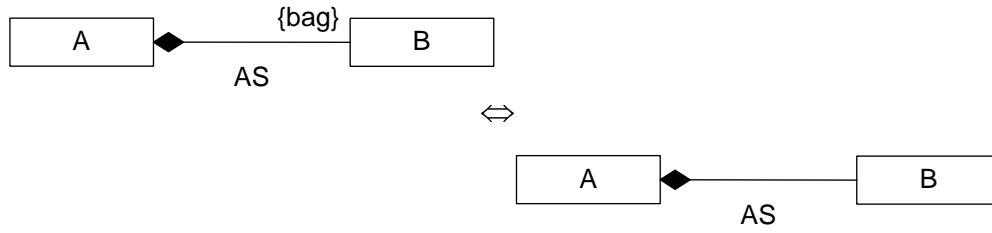
Jeżeli zachodzi równość powyższych zbiorów, wówczas $\mathcal{D}_1 = \mathcal{D}_2$. W przeciwnym wypadku spełnione są założenia twierdzenia 5.4.2 (str. 45). \square

Uwaga 6.4.1 (Liczności końców kompozycji)

Wartości większe niż *jeden*, należące do liczności końca po stronie kompozytu nie powodują niespójności semantycznej kompozycji, a są przejawem nieprecyzyjności opisu dziedziny (rys. 6.3). Niespójność semantyczną kompozycji powoduje natomiast brak wartości *jeden* w liczności tego końca (rys. 6.4). Wspomnianą niespójność wykazuje się w sposób podobny do pierwszej części dowodu twierdzenia 6.4.1. \square

Uwaga 6.4.2 (Kompozycja a unikalność powiązań)

Z twierdzeń 6.4.1 i 6.3.1 wynika, że powiązania należące do ekstensji kompozycji z oznaczeniem {bag} będą unikalne. Innymi słowy, oznaczenie {bag} w przypadku kompozycji nie zmienia semantyki diagramu (rys. 6.5). \square



Rysunek 6.5: Unikalność powiązań a kompozycja

6.5 Specjalizacja asocjacji w metamodelu UML

W przypadku specjalizowania asocjacji, metamodel UML nakłada tylko dwa warunki na diagram klas (por. punkt 8 definicji pojęciowego diagramu klas, str. 28):

- asocjacja oraz jej specjalizacja mają taką samą liczbę końców,
- asocjacja specjalizująca ma na swoich końcach te same klasyfikatory co asocjacja specjalizowana lub specjalizacje tych klasyfikatorów.

Powyższe warunki (należące do składni diagramu) odzwierciedlają częściowo semantykę asocjacji specjalizującej, tj. asocjacji, której powiązania są szczególnym przypadkiem powiązań asocjacji specjalizowanej. Pomędzy takimi asocjacjami istnieją jednak jeszcze inne — prezentowane dalej — zależności, które mogłyby zostać zapisane już na poziomie metamodelu UML.

6.6 Specjalizacja asocjacji a licznosc

Z przyjętej dla pojęciowego diagramu klas interpretacji związku specjalizacji, ekstensja asocjacji specjalizującej jest podzbiorem ekstensji asocjacji specjalizowanej. Wynika stąd, że dla danego zbioru obiektów, liczba łączących je powiązań, które należą do pierwszej z tych ekstensji, nie może być większa, niż liczba takich powiązań należących do drugiej. Formalnie ujmując to poniższe twierdzenie.

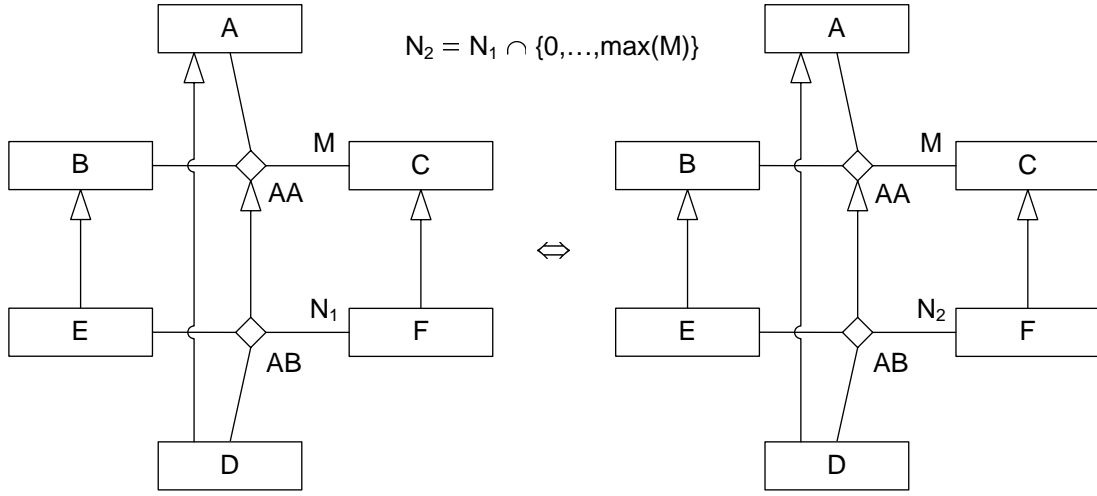
Twierdzenie 6.6.1 (Specjalizacja asocjacji a licznosc)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (6.22) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &\neq \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają asocjacje $AA, AB \in \mathcal{D}_1.\text{assocs}$, takie że AB jest specjalizacją AA :

$$AB \in \mathcal{D}_1.\text{specs}(AA), \quad (6.23)$$



Rysunek 6.6: Specjalizacja asocjacji a licznosc

przy czym pomiędzy licznosciami końca k asocjacji AB zachodzi na obu diagramach zależność²⁸:

$$\begin{aligned} \pi_k(\mathcal{D}_2.\text{mults}(AB)) = & \quad (6.24) \\ \pi_k(\mathcal{D}_1.\text{mults}(AB)) \cap \{0, \dots, \max(\pi_k(\mathcal{D}_1.\text{mults}(AA)))\}, & \end{aligned}$$

a licznosci pozostałych końców asocjacji AB oraz końców innych asocjacji nie różnią się na obu diagramach:

$$\bar{\pi}_k(\mathcal{D}_1.\text{mults}(AB)) = \bar{\pi}_k(\mathcal{D}_2.\text{mults}(AB)), \quad (6.25)$$

$$\forall as \in \mathcal{D}_1.\text{assocs} \setminus \{AB\} \cdot \mathcal{D}_1.\text{mults}(as) = \mathcal{D}_2.\text{mults}(as). \quad (6.26)$$

Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.6).

Dowód

Dowodzi się najpierw, że zachodzi wynikanie $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, a następnie $\mathcal{D}_2 \Rightarrow \mathcal{D}_1$.

$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $\text{Sat}(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 3 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $\text{Sat}(\mathcal{D}_1, \mathcal{S})$ i założenia (6.22).

Niech $as \in \mathcal{D}_2.\text{assocs}$, $i \in \{1, \dots, \text{len}(\mathcal{D}_2.\text{ends}(as))\}$ i $p \in \text{product}(as, i)$ (z założenia (6.22) i definicji funkcji product , ma ona jednakową postać dla diagramów \mathcal{D}_1 i \mathcal{D}_2). Jeżeli $as \neq AB$ lub $i \neq k$, to wspomniany punkt spełniony jest z $\text{Sat}(\mathcal{D}_1, \mathcal{S})$ oraz równań (6.25) i (6.26). Przyjmując dalej oznaczenia:

$$\alpha_{AA}(i, p) =_{\text{def}} |\{ln \in \mathcal{S}.\text{instances}(AA) : \bar{\pi}_i(\mathcal{S}.\text{ends}(ln)) = p\}| \quad (6.27)$$

$$\alpha_{AB}(i, p) =_{\text{def}} |\{ln \in \mathcal{S}.\text{instances}(AB) : \bar{\pi}_i(\mathcal{S}.\text{ends}(ln)) = p\}|, \quad (6.28)$$

²⁸Jeżeli X jest nieskończonym podzbiorem \mathbb{N} , to przyjmuje się $\{0, \dots, \max(X)\} =_{\text{def}} \mathbb{N}$.

pozostaje wykazać, że:

$$\alpha_{AB}(k, p) \in \pi_k(\mathcal{D}_2.\text{mults}(AB)). \quad (*)$$

Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, zatem:

$$\alpha_{AA}(k, p) \in \pi_k(\mathcal{D}_1.\text{mults}(AA)) \quad \text{i} \quad (6.29)$$

$$\alpha_{AB}(k, p) \in \pi_k(\mathcal{D}_1.\text{mults}(AB)), \quad (6.30)$$

a z założenia (6.23) i z punktu 7 definicji Sat (str. 33):

$$\mathcal{S}.\text{instances}(AB) \subseteq \mathcal{S}.\text{instances}(AA). \quad (6.31)$$

Zachodzi zatem nierówność:

$$\alpha_{AB}(k, p) \leq \alpha_{AA}(k, p) \quad (6.32)$$

i z równania (6.29):

$$\alpha_{AB}(k, p) \in \{0, \dots, \max(\pi_k(\mathcal{D}_1.\text{mults}(AA)))\}. \quad (6.33)$$

Z równania (6.30):

$$\alpha_{AB}(k, p) \in \quad (6.34)$$

$$\pi_k(\mathcal{D}_1.\text{mults}(AB)) \cap \{0, \dots, \max(\pi_k(\mathcal{D}_1.\text{mults}(AA)))\} \quad (6.35)$$

i własność (*) jest spełniona przy założeniu (6.24).

$\mathcal{D}_2 \Rightarrow \mathcal{D}_1$

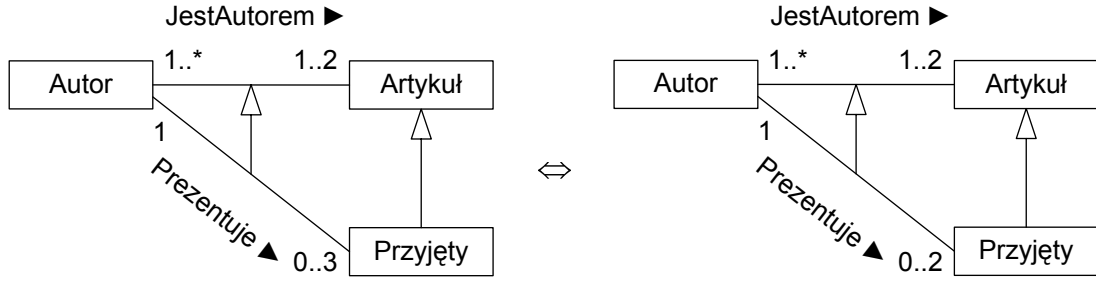
Z założeń twierdzenia:

$$\pi_k(\mathcal{D}_2.\text{mults}(AB)) \subseteq \pi_k(\mathcal{D}_1.\text{mults}(AB)). \quad (6.36)$$

Jeżeli zachodzi równość powyższych zbiorów, wówczas $\mathcal{D}_1 = \mathcal{D}_2$. W przeciwnym wypadku spełnione są założenia twierdzenia 5.4.2 (str. 45). \square

Przykład 6.6.1 (Nieprecyzyjna specyfikacja liczności)

Rysunek 6.7 ilustruje sytuację na konferencji naukowej. Uczestnicy mogą być autorami (lub współautorami) nie więcej niż dwóch zgłoszonych na konferencję artykułów. Jeżeli artykuł zostaje przyjęty na konferencję, jest prezentowany przez jednego z jego autorów. Na mocy twierdzenia 6.6.1, autor nie może mieć jednak więcej niż dwie prezentacje. \square



Rysunek 6.7: Nieprecyzyjna specyfikacja liczności

6.7 Specjalizacja asocjacji a unikalność powiązań

Jeżeli do ekstensji danej asocjacji mogą należeć tylko powiązania unikalne ze względu na łączone obiekty, to tym samym dotyczy to również asocjacji, które są jej specjalizacjami. Oznaczenie takich specjalizacji własnością $\{\text{bag}\}$ nie zmieni zatem semantyki diagramu.

Twierdzenie 6.7.1 (Specjalizacja asocjacji a unikalność powiązań)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (6.37) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &\neq \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają asocjacje $AA, AB \in \mathcal{D}_1.\text{assocs}$, takie że AB jest specjalizacją AA :

$$AB \in \mathcal{D}_1.\text{specs}(AA), \quad (6.38)$$

przy czym na diagramie \mathcal{D}_1 AB posiada własność $\{\text{bag}\}$, a na diagramie \mathcal{D}_2 nie posiada takiej własności. Pozostałe asocjacje nie różnią się na obu diagramach:

$$AB \in \mathcal{D}_1.\text{assocs}_{\text{bag}}, \quad (6.39)$$

$$\mathcal{D}_2.\text{assocs}_{\text{bag}} = \mathcal{D}_1.\text{assocs}_{\text{bag}} \setminus \{AB\}. \quad (6.40)$$

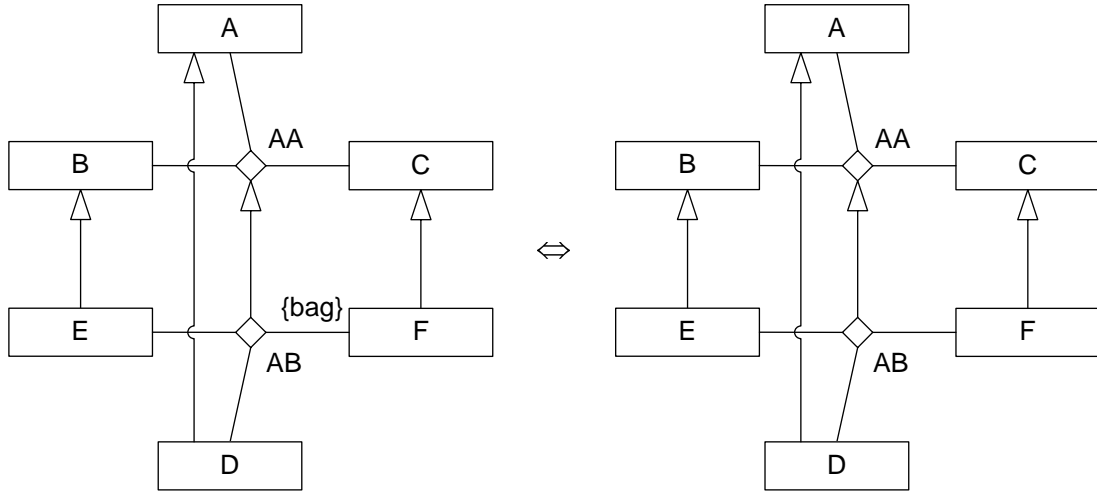
Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.8).

Dowód

Dowodzi się najpierw, że zachodzi wynikanie $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, a następnie $\mathcal{D}_2 \Rightarrow \mathcal{D}_1$.

$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $\text{Sat}(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie,



Rysunek 6.8: Specjalizacja a unikalność powiązań

że spełniony jest punkt 4 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (6.37).

Niech $as \in \mathcal{D}_2.\text{assocs} \setminus \mathcal{D}_2.\text{assocs}_{\text{bag}}$. Jeżeli $as \neq AB$, to $as \in \mathcal{D}_1.\text{assocs} \setminus \mathcal{D}_1.\text{assocs}_{\text{bag}}$ i wspomniany punkt spełniony jest z $Sat(\mathcal{D}_1, \mathcal{S})$. Pozostaje zatem wykazać własność:

$$\begin{aligned} \forall ln_1, ln_2 \in \mathcal{S}.\text{instances}(AB). & \quad (*) \\ ln_1 \neq ln_2 \Rightarrow \exists i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(AB))\}. & \\ \pi_i(\mathcal{S}.\text{ends}(ln_1)) \neq \pi_i(\mathcal{S}.\text{ends}(ln_2)). & \end{aligned}$$

Z założenia (6.38) oraz z punktu 8 definicji pojęciowego diagramu klas (str. 28):

$$\mathbf{len}(\mathcal{D}_1.\text{ends}(AA)) = \mathbf{len}(\mathcal{D}_1.\text{ends}(AB)). \quad (6.41)$$

Ponieważ zachodzi $Sat(\mathcal{D}_1, \mathcal{S})$, zatem:

$$\begin{aligned} \forall ln_1, ln_2 \in \mathcal{S}.\text{instances}(AA). & \quad (6.42) \\ ln_1 \neq ln_2 \Rightarrow \exists i \in \{1, \dots, \mathbf{len}(\mathcal{D}.\text{ends}(AA))\}. & \\ \pi_i(\mathcal{S}.\text{ends}(ln_1)) \neq \pi_i(\mathcal{S}.\text{ends}(ln_2)). & \end{aligned}$$

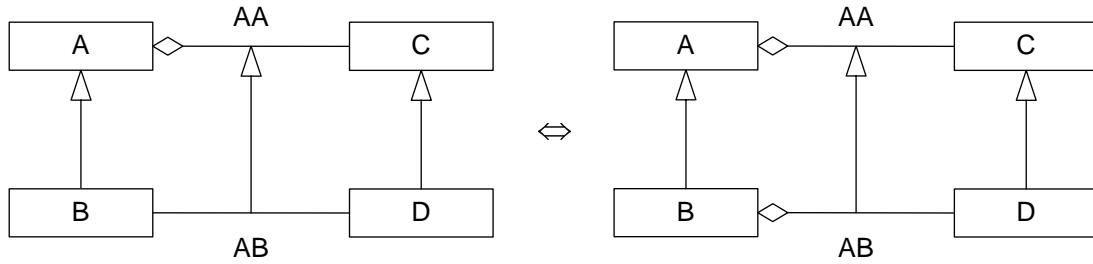
a z założenia (6.38) i z punktu 7 definicji Sat (str. 33):

$$\mathcal{S}.\text{instances}(AB) \subseteq \mathcal{S}.\text{instances}(AA). \quad (6.43)$$

Ostatecznie, z równań (6.41)–(6.43) wynika własność (*).

$\mathcal{D}_2 \Rightarrow \mathcal{D}_1$

Spełnione są założenia twierdzenia 5.4.3 (str. 46). \square



Rysunek 6.9: Specjalizacja agregacji

6.8 Specjalizacja agregacji

Jeżeli dana asocjacja jest agregacją, to — jak pokazuje poniższe twierdzenie — powiązania należące do ekstensji jej specjalizacji również spełniają warunki dla instancji agregacji. To oznacza, że specjalizacja agregacji jest semantycznie również agregacją, nawet jeżeli na diagramie występuje jako zwykła asocjacja²⁹.

Twierdzenie 6.8.1 (Specjalizacja agregacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &= \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &\neq \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (6.44) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają agregację $AA \in \mathcal{D}_1.\text{assocs}_{\text{agg}}$ oraz asocjację $AB \in \mathcal{D}_1.\text{assocs}$, taką że AB jest specjalizacją AA :

$$AB \in \mathcal{D}_1.\text{specs}(AA), \quad (6.45)$$

przy czym AB nie jest agregacją na diagramie \mathcal{D}_1 , a jest nią na diagramie \mathcal{D}_2 . Pozostałe asocjacje nie różnią się na obu diagramach:

$$AB \in \mathcal{D}_2.\text{assocs}_{\text{agg}} \quad (6.46)$$

$$\mathcal{D}_1.\text{assocs}_{\text{agg}} = \mathcal{D}_2.\text{assocs}_{\text{agg}} \setminus \{AB\}. \quad (6.47)$$

Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.9).

Dowód

Dowodzi się najpierw, że zachodzi wynikanie $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, a następnie $\mathcal{D}_2 \Rightarrow \mathcal{D}_1$.

²⁹Na brak odpowiedniego ograniczenia w metamodelu UML1, wymuszającego by specjalizacja agregacji była również agregacją, wskazuje (bez formalnego uzasadnienia) pozycja [Pons02]. Brak ten nadal występuje w metamodelu UML2.

$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 5 definicji Sat (str. 32). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (6.44).

Niech $parts_1$ i $parts_2$ wyznaczają części agregatu w stanie \mathcal{S} , odpowiednio dla diagramu \mathcal{D}_1 i \mathcal{D}_2 . Dalej dowodzi się nie wprost. Niech istnieje $ob \in Instances$, takie że:

$$ob \in parts_2(ob). \quad (6.48)$$

Rozpatruje się dwa przypadki:

1. Jeżeli ob jest swoją bezpośrednią częścią, wówczas istnieje agregacja $as \in \mathcal{D}_2.assoc_{agg}$ i powiązanie $ln \in \mathcal{S}.instances(as)$, dla którego:

$$ob = \pi_1(\mathcal{S}.ends(ln)) \wedge ob = \pi_2(\mathcal{S}.ends(ln)) \quad (6.49)$$

Gdyby $as \in \mathcal{D}_1.assoc_{agg}$, wówczas $ob \in parts_1(ob)$, co jest sprzeczne z $Sat(\mathcal{D}_1, \mathcal{S})$. A zatem $as \notin \mathcal{D}_1.assoc_{agg}$ i z założeń (6.46) i (6.47) $as = AB$. Z założenia (6.45) i z punktu 7 definicji $Sat(\mathcal{D}_1, \mathcal{S})$ (str. 33):

$$\mathcal{S}.instances(AB) \subseteq \mathcal{S}.instances(AA), \quad (6.50)$$

a stąd $ln \in \mathcal{S}.instances(AA)$. Zatem powiązanie ln jest instancją agregacji ze zbioru $\mathcal{D}_1.assoc_{agg}$, czyli $ob \in parts_1(ob)$, a to ponownie oznacza sprzeczność z $Sat(\mathcal{D}_1, \mathcal{S})$. Założenie (6.48) jest zatem niepoprawne.

2. Jeżeli ob jest swoją pośrednią częścią, wówczas istnieje n ($n \geq 2$) agregacji $as_1, \dots, as_n \in \mathcal{D}_2.assoc_{agg}$ oraz n powiązań:

$$ln_1 \in \mathcal{S}.instances(as_1), \dots, ln_n \in \mathcal{S}.instances(as_1), \quad (6.51)$$

dla których:

$$ob = \pi_1(\mathcal{S}.ends(ln_1)) \wedge ob = \pi_2(\mathcal{S}.ends(ln_n)) \wedge \\ \forall i \in \{1, \dots, n-1\} \cdot \pi_2(\mathcal{S}.ends(ln_i)) = \pi_1(\mathcal{S}.ends(ln_{i+1})).$$

Gdyby każda z tych agregacji należała do zbioru $\mathcal{D}_1.assoc_{agg}$, wówczas $ob \in parts_1(ob)$, co jest sprzeczne z $Sat(\mathcal{D}_1, \mathcal{S})$. Niech ciąg $as_{i_1}, \dots, as_{i_k}$ ($k, i_1, \dots, i_k \in \{1, \dots, n\}$, $i_j < i_l$ dla $j < l$) będzie najdłuższym podciągiem ciągu as_1, \dots, as_n , takim że:

$$\forall j \in \{1, \dots, k\} \cdot as_{i_j} \notin \mathcal{D}_1.assoc_{agg}. \quad (6.52)$$

Z założeń (6.46) i (6.47):

$$\forall j \in \{1, \dots, k\} \cdot as_{i_j} = AB. \quad (6.53)$$

Z założenia (6.45) i z punktu 7 definicji $Sat(\mathcal{D}_1, \mathcal{S})$ (str. 33):

$$\mathcal{S}.instances(AB) \subseteq \mathcal{S}.instances(AA), \quad (6.54)$$

a stąd:

$$\forall j \in \{1, \dots, k\} \cdot ln_{i_j} \in \mathcal{S}.instances(AA). \quad (6.55)$$

Zatem wszystkie powiązania ln_1, \dots, ln_n są instancjami agregacji ze zbioru $\mathcal{D}_1.assoc_{agg}$, czyli $ob \in parts_1(ob)$, co jest sprzeczne z $Sat(\mathcal{D}_1, \mathcal{S})$. Założenie (6.48) jest zatem niepoprawne.

$$\mathcal{D}_2 \Rightarrow \mathcal{D}_1$$

Spełnione są założenia twierdzenia 5.5.1 (str. 48). \square

6.9 Specjalizacja kompozycji

Podobnie jak w przypadku specjalizowania agregacji, obowiązujący dla kompozycji warunek o niewspółdzieleniu części również przenosi się na jej specjalizację. Innymi słowy, specjalizacja kompozycji jest semantycznie również kompozycją, nawet jeżeli nie jest to wyspecyfikowane wprost na diagramie.

Twierdzenie 6.9.1 (Specjalizacja kompozycji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned} \mathcal{D}_1.classes &= \mathcal{D}_2.classes, & \mathcal{D}_1.assoc_{agg} &= \mathcal{D}_2.assoc_{agg}, & (6.56) \\ \mathcal{D}_1.assoc &= \mathcal{D}_2.assoc, & \mathcal{D}_1.assoc_{com} &\neq \mathcal{D}_2.assoc_{com}, \\ \mathcal{D}_1.ends &= \mathcal{D}_2.ends, & \mathcal{D}_1.specs &= \mathcal{D}_2.specs, \\ \mathcal{D}_1.mults &= \mathcal{D}_2.mults, & \mathcal{D}_1.abstract &= \mathcal{D}_2.abstract \\ \mathcal{D}_1.assoc_{bag} &= \mathcal{D}_2.assoc_{bag}, \end{aligned}$$

i diagramy zawierają kompozycję $AA \in \mathcal{D}_1.assoc_{com}$ oraz agregację $AB \in \mathcal{D}_1.assoc_{agg}$, taką że AB jest specjalizacją AA :

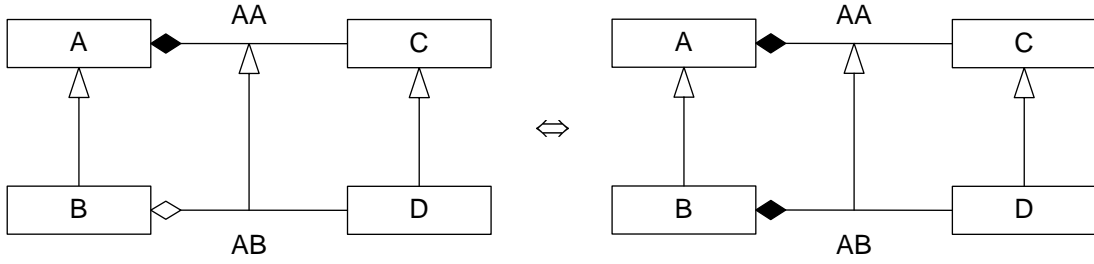
$$AB \in \mathcal{D}_1.specs(AA), \quad (6.57)$$

przy czym na diagramie \mathcal{D}_1 AB nie jest kompozycją, a jest nią na diagramie \mathcal{D}_2 . Pozostałe agregacje nie różnią się na obu diagramach:

$$AB \in \mathcal{D}_2.assoc_{com} \quad (6.58)$$

$$\mathcal{D}_1.assoc_{com} = \mathcal{D}_2.assoc_{com} \setminus \{AB\}. \quad (6.59)$$

Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.10).



Rysunek 6.10: Specjalizacja kompozycji

Dowód

Dowodzi się najpierw, że zachodzi wynikanie $\mathcal{D}_1 \Rightarrow \mathcal{D}_2$, a następnie $\mathcal{D}_2 \Rightarrow \mathcal{D}_1$.

$\mathcal{D}_1 \Rightarrow \mathcal{D}_2$

Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}_1)$. W ramach dowodu, że zachodzi $Sat(\mathcal{D}_2, \mathcal{S})$ pokazane zostanie, że spełniony jest punkt 6 definicji Sat (str. 33). Spełnienie pozostałych punktów tej definicji wynika bezpośrednio z $Sat(\mathcal{D}_1, \mathcal{S})$ i założenia (6.56).

Dowodzi się nie wprost. Niech istnieją kompozycje $as_1, as_2 \in \mathcal{D}_2.\text{assoc}_{\text{com}}$ oraz powiązania $ln_1 \in \mathcal{S}.\text{instances}(as_1)$ i $ln_2 \in \mathcal{S}.\text{instances}(as_2)$, takie że:

$$ln_1 \neq ln_2 \wedge \pi_2(\mathcal{S}.\text{ends}(ln_1)) = \pi_2(\mathcal{S}.\text{ends}(ln_2)). \quad (6.60)$$

Gdyby obie kompozycje należały do zbioru $\mathcal{D}_1.\text{assoc}_{\text{com}}$, byłyby to sprzeczne z $Sat(\mathcal{D}_1, \mathcal{S})$. Zachodzi zatem jeden z trzech przypadków:

$$(1) \quad as_1 \in \mathcal{D}_1.\text{assoc}_{\text{com}} \wedge as_2 \notin \mathcal{D}_1.\text{assoc}_{\text{com}}, \quad (6.61)$$

$$(2) \quad as_1 \notin \mathcal{D}_1.\text{assoc}_{\text{com}} \wedge as_2 \in \mathcal{D}_1.\text{assoc}_{\text{com}}, \quad (6.62)$$

$$(3) \quad as_1 \notin \mathcal{D}_1.\text{assoc}_{\text{com}} \wedge as_2 \notin \mathcal{D}_1.\text{assoc}_{\text{com}}. \quad (6.63)$$

Dalej, tak jak w dowodzie twierdzenia 6.8.1, wykazuje się że:

$$as_i \notin \mathcal{D}_1.\text{assoc}_{\text{com}} \Rightarrow as_i = AB \quad (6.64)$$

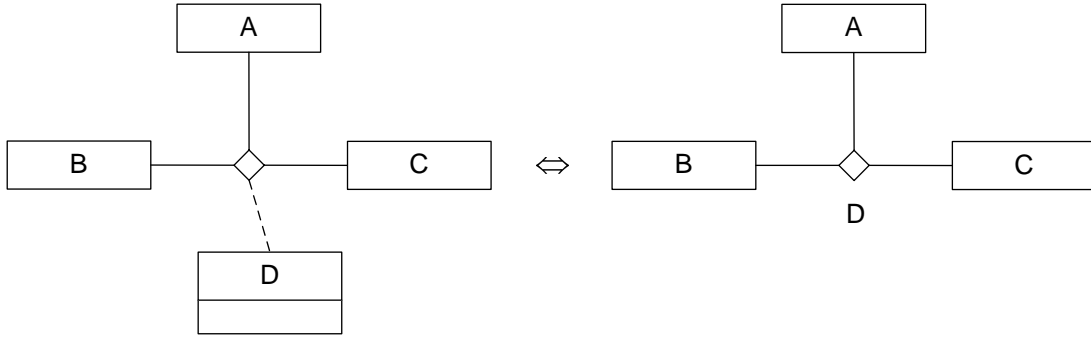
i oba powiązania ln_1 i ln_2 są instancjami kompozycji ze zbioru $\mathcal{D}_1.\text{assoc}_{\text{com}}$, co jest sprzeczne z $Sat(\mathcal{D}_1, \mathcal{S})$.

$\mathcal{D}_2 \Rightarrow \mathcal{D}_1$

Spełnione są założenia twierdzenia 5.5.2 (str. 50). \square

6.10 „Pusta” klasa asocjacji

Klasa asocjacji, poza tym, że łączy pewien zbiór klasyfikatorów może posiadać własności typowe dla klasy, tj. może brać udział w asocjacjach, w tym posiadać atrybuty. Jeżeli jednak takich własności na diagramie nie posiada, zatem nie powinna się semantycznie różnić od zwykłej asocjacji. Wyraża to poniższe twierdzenie.



Rysunek 6.11: Równoważność klasy asocjacji i asocjacji

Twierdzenie 6.10.1 (Równoważność klasy asocjacji i asocjacji)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$, takie że:

$$\begin{aligned}
 \mathcal{D}_1.\text{classes} &\neq \mathcal{D}_2.\text{classes}, & \mathcal{D}_1.\text{assocs}_{\text{agg}} &= \mathcal{D}_2.\text{assocs}_{\text{agg}}, & (6.65) \\
 \mathcal{D}_1.\text{assocs} &= \mathcal{D}_2.\text{assocs}, & \mathcal{D}_1.\text{assocs}_{\text{com}} &= \mathcal{D}_2.\text{assocs}_{\text{com}}, \\
 \mathcal{D}_1.\text{ends} &= \mathcal{D}_2.\text{ends}, & \mathcal{D}_1.\text{specs} &= \mathcal{D}_2.\text{specs}, \\
 \mathcal{D}_1.\text{mults} &= \mathcal{D}_2.\text{mults}, & \mathcal{D}_1.\text{abstract} &= \mathcal{D}_2.\text{abstract} \\
 \mathcal{D}_1.\text{assocs}_{\text{bag}} &= \mathcal{D}_2.\text{assocs}_{\text{bag}},
 \end{aligned}$$

i diagramy zawierają klasyfikator $D \in \mathcal{D}_1.\text{assocs}$, który jest klasą asocjacji na diagramie \mathcal{D}_1 , a asocjacją na diagramie \mathcal{D}_2 . Typy pozostałych klasyfikatorów nie różnią się na obu diagramach:

$$D \in \mathcal{D}_1.\text{classes}, \tag{6.66}$$

$$\mathcal{D}_2.\text{classes} = \mathcal{D}_1.\text{classes} \setminus \{D\}. \tag{6.67}$$

Wówczas diagram \mathcal{D}_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 (rys. 6.11)³⁰.

Dowód

Niech $\mathcal{S} \in \text{States}$. Dowód, że zachodzi $\text{Sat}(\mathcal{D}_1, \mathcal{S}) \Leftrightarrow \text{Sat}(\mathcal{D}_2, \mathcal{S})$ ogranicza się dalej do punktu 1 definicji Sat (str. 32). Równoważność w przypadku pozostałych punktów tej definicji wynika bezpośrednio z założenia (6.65).

Z założeń twierdzenia:

$$\mathcal{D}_1.\text{classifiers} = \mathcal{D}_2.\text{classifiers}, \tag{6.68}$$

a zatem:

$$\begin{aligned}
 \mathcal{D}_1.\text{classifiers} &\subseteq \text{dom}(\mathcal{S}.\text{instances}) \Leftrightarrow \\
 \mathcal{D}_2.\text{classifiers} &\subseteq \text{dom}(\mathcal{S}.\text{instances}). \quad \square
 \end{aligned}
 \tag{6.69}$$

³⁰Z założeń twierdzenia wynika, że na diagramie \mathcal{D}_1 , klasa asocjacji D nie jest połączona żadną asocjacją (w tym nie ma atrybutów) oraz może specjalizować jedynie asocjacje — por. punkt 8 definicji pojęciowego diagramu klas (str. 28).

Rozdział 7

Semantyka ograniczeń

Niniejszy rozdział jest propozycją formalnego wyrażania ograniczeń w modelowaniu pojęciowym. Pokazuje, w jaki sposób zapisywać formalnie niektóre z powszechnie używanych ograniczeń, a tym samym, jak rozszerzać zdefiniowany wcześniej pojęciowy diagramu klas o nowe elementy modelowania.

7.1 Ograniczenie

Ograniczenie w UML jest semantyczną restrykcją nałożoną na element diagramu. W niniejszym rozdziale definiuje się ograniczenie w sposób bardziej ogólny, jako warunek niekoniecznie dotyczący któregoś z elementów diagramu. Z drugiej strony, ograniczenie zawęża się tu do przypadku pojęciowego diagramu klas, traktując je jako warunek nałożony *de facto* na stan dziedziny.

Należy podkreślić, że graficzne elementy diagramu same w sobie są pewnym zapisem ograniczeń, np. determinują możliwe powiązania i ich liczbę. Nie pozwalają jednak na wyrażanie dowolnie złożonych warunków.

Definicja 7.1.1 (Ograniczenie)

Ograniczeniem jest dowolny częściowy predykat nad zbiorem States, tj. funkcja boolowska:

$$\Delta: \text{States} \rightarrow \mathbb{B}.$$

Jeżeli zachodzi $\Delta(\mathcal{S})$ (tzn. $\Delta(\mathcal{S}) = T$), to mówi się, że ograniczenie Δ jest *spełnione w stanie* \mathcal{S} . \square

Definicja 7.1.2 (Zbiór Constraints)

Przez Constraints oznacza się zbiór wszystkich ograniczeń postaci jak w definicji 7.1.1, czyli:

$$\text{Constraints} =_{def} \text{States} \rightarrow \mathbb{B}. \quad \square$$

7.2 Znaczenie ograniczenia

Jako interpretację ograniczenia przyjmuje się zbiór stanów dziedziny, w których ograniczenie jest spełnione. Formalnie ujmując to poniższa definicja.

Definicja 7.2.1 (Znaczenie ograniczenia)

Niech $\Delta \in \text{Constraints}$. Znaczeniem (interpretacją) ograniczenia Δ jest zbiór:

$$\mathcal{M}(\Delta),$$

gdzie \mathcal{M} jest funkcją postaci³¹:

$$\begin{aligned} \mathcal{M}: \text{Constraints} &\rightarrow \mathcal{P}(\text{States}), \\ \mathcal{M}(\Delta) &=_{\text{def}} \{ \mathcal{S} \in \text{States} : \Delta(\mathcal{S}) \}. \quad \square \end{aligned}$$

7.3 Diagram z ograniczeniem

Dalej definiuje się formalnie pojęciowy diagram klas z ograniczeniem. Naturalną interpretacją takiego diagramu są stany, w których spełnione są zarówno warunki wynikające z graficznych elementów diagramu, jak również te zapisane ograniczeniem.

Definicja 7.3.1 (Pojęciowy diagram klas z ograniczeniem)

Jeżeli $\mathcal{D} \in \text{Diagrams}$ i $\Delta \in \text{Constraints}$, to uporządkowaną dwójkę:

$$(\mathcal{D}, \Delta)$$

nazywa się *pojęciowym diagramem klas z ograniczeniem*. \square

Definicja 7.3.2 (Znaczenie diagramu*)

Niech $\mathcal{D} \in \text{Diagrams}$ i $\Delta \in \text{Constraints}$. Znaczeniem (interpretacją) diagramu \mathcal{D} z ograniczeniem Δ jest zbiór:

$$\mathcal{M}(\mathcal{D}, \Delta),$$

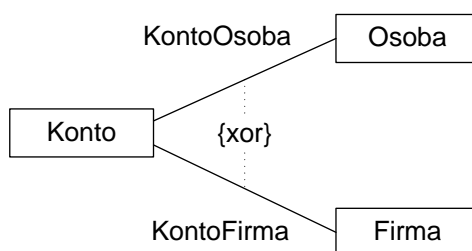
gdzie \mathcal{M} jest funkcją postaci:

$$\begin{aligned} \mathcal{M}: \text{Diagrams} \times \text{Constraints} &\rightarrow \mathcal{P}(\text{States}), \\ \mathcal{M}(\mathcal{D}, \Delta) &=_{\text{def}} \mathcal{M}(\mathcal{D}) \cap \mathcal{M}(\Delta). \quad \square \end{aligned}$$

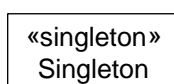
Uwaga 7.3.1 (Diagram bez ograniczenia)

Powyższe definicje obejmują również przypadek diagramu pojęciowego bez ograniczenia, zdefiniowanego w rozdziale 3. Miejsce brakującego ograniczenia może bowiem zająć dowolna tautologia. \square

³¹Ze względu na inną dziedzinę funkcji, ponowne użycie symbolu \mathcal{M} nie będzie dalej prowadzić do niejednoznaczności z definicją znaczenia diagramu.



Rysunek 7.1: Ograniczenie {xor}



Rysunek 7.2: Stereotyp «singleton»

Przykład 7.3.1 (Ograniczenie {xor})

Jednym z predefiniowanych w UML ograniczeń jest ograniczenie {xor}. Na rysunku 7.1 łączy ono dwie asocjacje, reprezentując warunek, że dla danego obiektu klasy *Konto* tylko jedna z asocjacji może posiadać w danej chwili powiązania. Przyjmując, że:

$$\begin{aligned} \mathcal{D}.ends(KontoOsoba) &= [Konto, Osoba] \text{ i} \\ \mathcal{D}.ends(KontoFirma) &= [Konto, Firma], \end{aligned}$$

odpowiada to formalnemu ograniczeniu:

$$\begin{aligned} \Delta(\mathcal{S}) =_{def} \forall ln_1 \in \mathcal{S}.instances(KontoOsoba) \cdot \\ \forall ln_2 \in \mathcal{S}.instances(KontoFirma) \cdot \\ \pi_1(\mathcal{S}.ends(ln_1)) \neq \pi_1(\mathcal{S}.ends(ln_2)). \quad \square \end{aligned}$$

Przykład 7.3.2 (Stereotyp «singleton»)

Czasem w modelowaniu zachodzi potrzeba wprowadzenia ograniczenia, by dla danej klasy istniała dokładnie jedna instancja [Gamma05]. Zazwyczaj wyraża się to na diagramie przy użyciu stereotypu «singleton»³². Na diagramie z rysunku 7.2 odpowiada on formalnemu ograniczeniu:

$$\Delta(\mathcal{S}) =_{def} |\mathcal{S}.instances(Singleton)| = 1. \quad \square \quad (7.1)$$

³²Inny sposób oznaczenia klasy o dokładnie jednej instancji to umieszczenie cyfry „1” w prawym górnym rogu prostokąta reprezentującego klasę [Larman01].

7.4 Ograniczenia zbioru generalizacji

W skład *zbioru generalizacji*³³ wchodzi związki specjalizacji tworzące jeden aspekt specjalizacji danego klasyfikatora. Wszystkie związki w zbiorze generalizacji łączą zatem tego samego przodka w hierarchii specjalizacji. Zbiór generalizacji może być oznaczony jedną z czterech par ograniczeń:

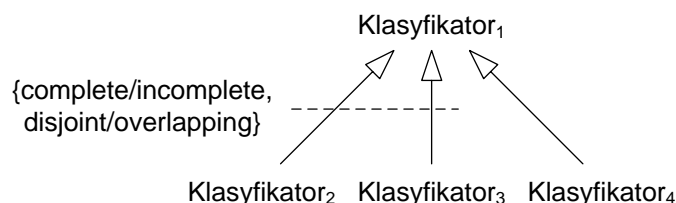
1. {complete, disjoint},
2. {incomplete, disjoint},
3. {complete, overlapping},
4. {incomplete, overlapping},

przy czym oznaczenie {incomplete, disjoint} jest oznaczeniem domyślnym. Nieformalna interpretacja poszczególnych ograniczeń jest następująca:

- *complete* — klasyfikatory specjalizujące całkowicie pokrywają aspekt specjalizacji; każda instancja klasyfikatora specjalizowanego jest jednocześnie instancją któregoś z klasyfikatorów specjalizujących.
- *incomplete* — klasyfikatory specjalizujące nie pokrywają całkowicie aspektu specjalizacji; instancja klasyfikatora specjalizowanego może nie być instancją żadnego z klasyfikatorów specjalizujących.
- *disjoint* — klasyfikatory specjalizujące wzajemnie się wykluczają; żadna instancja nie może być jednocześnie instancją więcej niż jednego spośród tych klasyfikatorów.
- *overlapping* — klasyfikatory specjalizujące nie wykluczają się wzajemnie; dana instancja może być jednocześnie instancją więcej niż jednego spośród tych klasyfikatorów.

Dla zbioru generalizacji z rysunku 7.3, poszczególne ograniczenia zapisano poniżej formalnie:

³³Zbiór generalizacji wprowadzono w UML2; zastępuje on pojęcie *diskryminatora* z UML1.



Rysunek 7.3: Zbiór generalizacji

- *complete*:

$$\begin{aligned} \mathcal{S}.\text{instances}(\text{Klasyfikator}_1) = \\ \mathcal{S}.\text{instances}(\text{Klasyfikator}_2) \cup \mathcal{S}.\text{instances}(\text{Klasyfikator}_3), \end{aligned}$$

- *incomplete* — brak dodatkowego ograniczenia,
- *disjoint*:

$$\mathcal{S}.\text{instances}(\text{Klasyfikator}_2) \cap \mathcal{S}.\text{instances}(\text{Klasyfikator}_3) = \emptyset,$$

- *overlapping* — brak dodatkowego ograniczenia.

Warto zauważyć, że ograniczenie {complete} oznacza abstrakcyjność klasyfikatora Klasyfikator_1 .

7.5 Spójność a ograniczenia

Przedstawione w rozdziale 4 zagadnienie spójności semantycznej pojęciowego diagramu klas w naturalny sposób rozszerza się na przypadek diagramu z ograniczeniem, prowadząc do poniższych definicji.

Definicja 7.5.1 (Spójność klasyfikatora*)

Niech $\mathcal{D} \in \text{Diagrams}$, $\Delta \in \text{Constraints}$ i $cf \in \text{Classifiers}$. Klasyfikator cf jest *spójny semantycznie w kontekście diagramu \mathcal{D} z ograniczeniem Δ* wtedy i tylko wtedy, gdy:

$$\exists \mathcal{S} \in \mathcal{M}(\mathcal{D}, \Delta) \cdot \mathcal{S}.\text{instances}(cf) \neq \emptyset.$$

W przeciwnym wypadku mówi się, że klasyfikator jest *niespójny semantycznie*. \square

Definicja 7.5.2 (Spójność diagramu*)

Diagram $\mathcal{D} \in \text{Diagrams}$ z ograniczeniem $\Delta \in \text{Constraints}$ jest *spójny semantycznie* wtedy i tylko wtedy, gdy dowolny klasyfikator $cf \in \text{Classifiers}$ jest spójny semantycznie w ich kontekście. W przeciwnym wypadku mówi się, że diagram jest *niespójny semantycznie*. \square

Przykład 7.5.1 (Niespójna specjalizacja wielokrotna)

Niech (\mathcal{D}, Δ) będzie formalną reprezentacją diagramu z rysunku 7.4, gdzie:

$$\Delta(\mathcal{S}) =_{\text{def}} \mathcal{S}.\text{instances}(B) \cap \mathcal{S}.\text{instances}(C) = \emptyset. \quad (7.2)$$

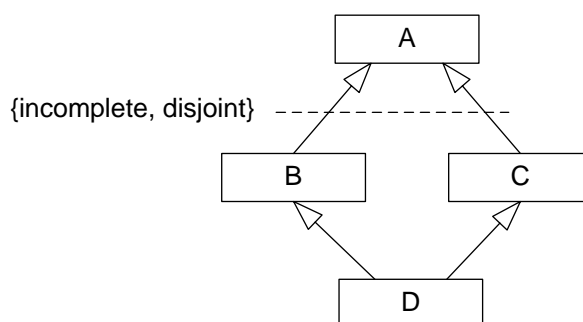
Niech $\mathcal{S} \in \mathcal{M}(\mathcal{D}, \Delta)$. Z punktu 7 definicji *Sat* (str. 33) zachodzi:

$$\mathcal{S}.\text{instances}(D) \subseteq \mathcal{S}.\text{instances}(B) \cap \mathcal{S}.\text{instances}(C), \quad (7.3)$$

a ponieważ ograniczenie Δ jest spełnione w stanie \mathcal{S} :

$$\mathcal{S}.\text{instances}(D) = \emptyset, \quad (7.4)$$

co oznacza niespójność semantyczną klasy D . \square



Rysunek 7.4: Niespójna specjalizacja wielokrotna

7.6 Wynikanie z ograniczeniami

Podobnie jak definicje dotyczące spójności semantycznej, definicje wynikania i równoważności semantycznej również dają się w prosty sposób rozszerzyć na przypadek diagramu z ograniczeniem. Wprowadza się je poniżej.

Definicja 7.6.1 (Wynikanie*)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$ i $\Delta_1, \Delta_2 \in \text{Constraints}$. Diagram \mathcal{D}_2 z ograniczeniem Δ_2 wynika semantycznie z diagramu \mathcal{D}_1 z ograniczeniem Δ_1 , co zapisuje się:

$$(\mathcal{D}_1, \Delta_1) \Rightarrow (\mathcal{D}_2, \Delta_2),$$

wtedy i tylko wtedy, gdy:

$$\mathcal{M}(\mathcal{D}_1, \Delta_1) \subseteq \mathcal{M}(\mathcal{D}_2, \Delta_2),$$

czyli w każdym stanie, w którym spełnione są \mathcal{D}_1 i Δ_1 spełnione są również \mathcal{D}_2 i Δ_2 . \square

Definicja 7.6.2 (Równoważność*)

Niech $\mathcal{D}_1, \mathcal{D}_2 \in \text{Diagrams}$ i $\Delta_1, \Delta_2 \in \text{Constraints}$. Diagram \mathcal{D}_1 z ograniczeniem Δ_1 jest równoważny semantycznie diagramowi \mathcal{D}_2 z ograniczeniem Δ_2 , co zapisuje się:

$$(\mathcal{D}_1, \Delta_1) \Leftrightarrow (\mathcal{D}_2, \Delta_2),$$

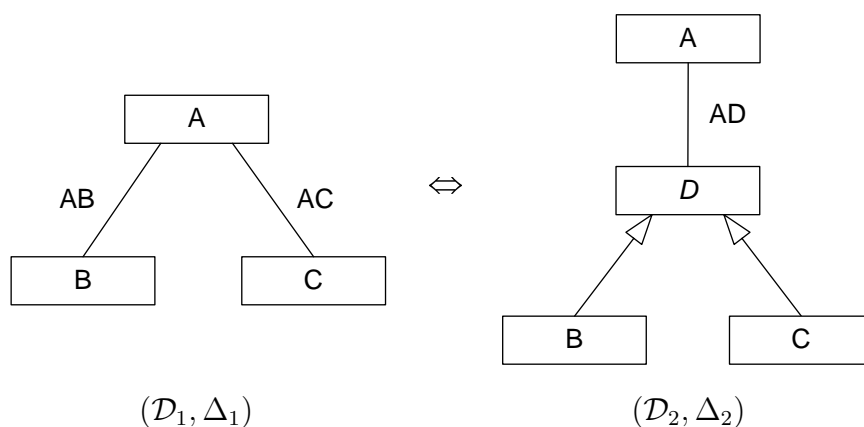
wtedy i tylko wtedy, gdy:

$$\mathcal{M}(\mathcal{D}_1, \Delta_1) = \mathcal{M}(\mathcal{D}_2, \Delta_2),$$

czyli \mathcal{D}_1 i Δ_1 są spełnione w danym stanie, wtedy i tylko wtedy, gdy spełnione są w nim również \mathcal{D}_2 i Δ_2 . \square

Propozycja 7.6.1 (Dokumentacja przekształceń diagramów)

Relacje wynikania i równoważności semantycznej dla pojęciowego diagramu klas



Rysunek 7.5: Wynikanie z ograniczeniami

z ograniczeniem pozwalają zapisywać przekształcenia pomiędzy diagramami o dowolnie różnych zbiorach klasyfikatorów. Semantyka klasyfikatora brakującego na danym diagramie może być zapisana w postaci odpowiedniego ograniczenia. Pokazuje to rysunek 7.5, gdzie:

$$\Delta_1(\mathcal{S}) =_{def} \mathcal{S}.instances(D) = \mathcal{S}.instances(B) \cup \mathcal{S}.instances(C) \wedge \\ \mathcal{S}.instances(AD) = \mathcal{S}.instances(AB) \cup \mathcal{S}.instances(AC)$$

oraz

$$\Delta_2(\mathcal{S}) =_{def} \mathcal{S}.instances(AD) = \mathcal{S}.instances(AB) \cup \mathcal{S}.instances(AC).$$

Tak udokumentowane przekształcenia są całkowicie weryfikowalne. \square

Rozdział 8

Podsumowanie

W pracy zaprezentowano rozwiązanie problemu formalizacji semantyki i wnioskowania na temat diagramu klas w UML, rysowanego z perspektywy pojęciowej. Na proponowane rozwiązanie składają się opracowane przez autora następujące elementy:

- Formalny opis składni i semantyki diagramu klas.
- Definicje spójności, wynikania oraz równoważności semantycznej diagramów.
- Ujęte w formie twierdzeń podstawowe reguły wynikania.

Opracowany opis formalny diagramu klas cechuje się kompletnością i aktualnością:

- Uwzględniono elementy dotąd pomijane lub formalizowane niekompletnie, takie jak asocjacja o arności większej niż dwa, agregacja i kompozycja, oraz całkowicie nowe w tego typu pracach, jak specjalizacja asocjacji i asocjacja o nieunikalnych powiązaniach.
- Przy formalizacji diagramu uwzględniono zmiany wprowadzone w wersji 2.0 standardu UML.

Uzyskane w pracy wyniki umożliwiają wykrywanie błędów i braków w standardzie UML. W pracy zidentyfikowano dwa obszary niedoskonałości tego standardu:

- W oparciu o definicję spójności semantycznej wskazano niebezpieczeństwo, polegające na możliwości utworzenia poprawnego syntaktycznie diagramu, reprezentującego sprzeczne (nieimplementowalne) własności. Wykazano między innymi niespójność semantyczną w przypadku prostej agregacji rekursywnej, a następnie, wykorzystując reguły wynikania i ich udowodnione własności, niespójności dla przypadków złożonej agregacji rekursywnej.
- W oparciu o definicję równoważności semantycznej diagramów wskazano na możliwość tworzenia konstrukcji, których pewne elementy reprezentują w efekcie silniejsze ograniczenia niż wynikałoby to tylko z użytych oznaczeń.

Tego typu konstrukcje na diagramach charakteryzują się wysokim ryzykiem błędnej interpretacji, co pokazano między innymi na przykładzie związku specjalizacji asocjacji.

Wyniki pracy mogą zatem posłużyć do wprowadzenia zmian i ulepszeń w przyszłych wersjach standardu UML.

Zaprezentowany w pracy aparat formalny dotyczący diagramu klas tworzy podstawę do:

- kontroli spójności semantycznej diagramów,
- przekształcania i porównywania diagramów.

Stwarza to możliwości weryfikacji modeli pojęciowych oraz procesu ich konstrukcji. Szersze zastosowanie rezultatów pracy wymaga w tym przypadku systematyzacji, bądź całkowitej automatyzacji procesu wnioskowania, tak by uniknąć żmudnych i podatnych na błędy dowodów matematycznych. Zagadnienia te pozostały poza ramami niniejszej pracy i wydają się najbardziej interesującym i pożądanym kierunkiem jej rozwoju.

Bibliografia

- [Andrews91] Derek Andrews, Darrel C. Ince. *Practical Formal Methods with VDM*. Mc-GrawHill, London, 1991.
- [Aredo99] Demissie B. Aredo, Issa Traoré, Ketil Stølen. *Towards a Formalization of UML Class Structure in PVS*. Research Report 272, Department of Informatics, University of Oslo, 1999.
- [Barbier03] Franck Barbier, Brian Henderson-Sellers, Annig Le Parc-Lacayrelle, Jean-Michel Bruel. Formalization of the Whole-Part Relationship in the Unified Modeling Language. *IEEE Transactions on Software Engineering*, Vol. 29, No. 5, pp. 459–470, 2003.
- [Berardi03] Daniela Berardi, Andrea Calì, Diego Calvanese, Giuseppe De Giacomo. *Reasoning on UML Class Diagrams*. Technical Report, Dipartimento di Informatica e Sistemistica, Università di Roma „La Sapienza”, 2003.
- [Booch94] Grady Booch. *Object-Oriented Analysis and Design with Applications*. Benjamin/Cummings, Redwood City, Calif., 2nd ed., 1994.
- [Booch01] Grady Booch, James Rumbaugh, Ivar Jacobson. *UML przewodnik użytkownika*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2001.
- [Bruel01] Jean-Michel Bruel, Brian Henderson-Sellers, Franck Barbier, Annig Le Parc, Robert B. France. Improving the UML Metamodel to Rigorously Specify Aggregation and Composition. *Proceedings of the 7th Int. Conf. on Object-Oriented Information Systems (OOIS'01)*, pp. 5–14, Springer-Verlag, 2001.
- [Calì02] Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini. A Formal Framework for Reasoning on UML Class Diagrams. *Proc. of the 13th Int. Sym. on Methodologies for Intelligent Systems (ISMIS 2002)*, 2002.
- [Chen76] Peter P. Chen. The Entity-Relationship Model – Toward a Unified View of Data. *ACM Transactions on Database Systems*, Vol. 1, No. 1, pp. 9–36, 1976.

-
- [Clark04] Tony Clark, Andy Evans, Paul Sammut, James Willans. *Applied Metamodelling: A Foundation for Language Driven Development*. Xactium, 2004. Internet: www.xactium.com.
- [Coad91a] Peter Coad, Edward Yourdon. *Object-Oriented Analysis*. Prentice Hall, Englewood Cliffs, N.J., 2nd ed., 1991.
- [Coad91b] Peter Coad, Edward Yourdon. *Object-Oriented Design*. Prentice Hall, Englewood Cliffs, N.J., 1991.
- [Crow95] Judy Crow, Sam Owre, John Rushby, Natarajan Shankar, Mandayam Srivas. A Tutorial Introduction to PVS. *Workshop on Industrial-Strength Formal Specification Techniques*, Boca Raton, Florida, 1995.
- [DeMarco79] Tom DeMarco. *Structured Analysis and System Specification*. Yourdon Press, Englewood Cliffs, 1979.
- [Egyed00] Alexander Egyed. Semantic Abstraction Rules for Class Diagrams. *Proc. of the 15th IEEE Int. Conf. on Automated Software Engineering (ASE)*, pp. 301–304, 2000.
- [Egyed02] Alexander Egyed. Automated Abstraction of Class Diagrams. *ACM Transactions on Software Engineering and Methodology*, Vol. 11, No. 4, pp. 449–491, 2002.
- [Evans98a] Andy Evans, Jean M. Bruel, Robert B. France, Kevin Lano, Bernhard Rumpe. Making UML Precise. *Proceedings of the OOP-SLA '98 Workshop on Formalizing UML*, 1998.
- [Evans98b] Andy Evans, Robert B. France, Kevin Lano, Bernhard Rumpe. The UML as a Formal Modeling Notation. *UML '98: Beyond the Notation, First International Workshop*, Mulhouse, France, pp. 336–348, 1998.
- [Evans98c] Andy Evans. Reasoning with UML Class Diagrams. *Second IEEE Workshop on Industrial Strength Formal Specification Techniques (WIFT'98)*, 1998.
- [Flasiński97] Mariusz Flasiński. *Wstęp do analitycznych metod projektowania systemów informatycznych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1997.
- [Fowler97] Martin Fowler. *Analysis Patterns: Reusable Object Models*. Addison-Wesley, MA, 1997.
- [Fowler02] Martin Fowler, Kendall Scott. *UML w kropelce*. Oficyna Wydawnicza LTP, Warszawa, 2002.

- [Fowler04] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, Boston, 3rd ed., 2004.
- [France99] Robert B. France. A Problem-Oriented Analysis of Basic UML Static Requirements Modeling Concepts. *OOPSLA '99: Proceedings of the 14th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pp. 57–69, 1999.
- [Funes03] Anna Maria Funes, Chris George. Formalizing UML Class Diagrams. In: Liliana Favre (Ed.), *UML and the Unified Process*, pp. 129–198, Idea Group Publishing, 2003.
- [Gamma05] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. *Wzorce projektowe: Elementy oprogramowania obiektowego wielokrotnego użytku*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2005.
- [Genova02] Gonzalo Génova, Juan Llorens, Paloma Martínez. The Meaning of Multiplicity of N-ary Associations in UML. *Software and Systems Modeling*, Vol. 2, No. 2, pp. 86–97, 2002.
- [George92] Chris George, Peter Haff, Anne E. Haxthausen, Robert Milne. *The RAISE Specification Language*. BCS Practitioner Series, Prentice Hall, 1992.
- [Gogolla98] Martin Gogolla, Mark Richters. Equivalence Rules for UML Class Diagrams. *UML'98: Beyond the Notation, First International Workshop*, Mulhouse, France, pp. 87–96, 1998.
- [Górski02] Janusz Górski, Aleksander Jarzębowicz. Wykrywanie anomalii w modelach obiektowych za pomocą metody UML-HAZOP. *Materiały IV Krajowej Konferencji Inżynierii Oprogramowania*, Tarnowo Podgórne k. Poznania, pp. 125–140, 2002.
- [Harel00] David Harel, Bernhard Rumpe. *Modeling Languages: Syntax, Semantics and All That Stuff – Part I: The Basic Stuff*. Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, Israel, No. MCS00-16, 2000.
- [Hatley88] Derek J. Hatley, Imtiaz A. Pirbhai. *Strategies for Real-Time System Specification*. Dorset House, New York, 1988.
- [Hnatkowska04] Bogumiła Hnatkowska, Zbigniew Huzar, Lech Tuzinkiewicz. Interpretacja mnogościowa związków pomiędzy klasami. In: Janusz Górski, Andrzej Wardziński (Eds.), *Inżynieria oprogramowania — Nowe wyzwania*, Wydawnictwa Naukowo-Techniczne, Warszawa, 2004.

-
- [ISO-VDM] ISO. *VDM-SL First Committee Draft Standard*. ISO/IEC JTC1/SC22/WG19 N-20, 1993.
- [Jacobson92] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, Wokingham, 1992.
- [Jones90] Cliff B. Jones. *Systematic Software Development Using VDM*. Prentice Hall International, 2nd ed., 1990.
- [Kroll03] Per Kroll, Philippe Kruchten. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley, Boston, 2003.
- [Kuratowski80] Kazimierz Kuratowski. *Wstęp do teorii mnogości i topologii*. Państwowe Wydawnictwo Naukowe, Warszawa, 1980.
- [Larman01] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall, Englewood Cliffs, N.J., 2nd ed., 2001.
- [Martin91] James Martin, James J. Odell. *Object-Oriented Analysis and Design*. Prentice Hall, Englewood Cliffs, N.J., 1991.
- [Martin97] James Martin, James J. Odell. *Podstawy metod obiektowych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1997.
- [Miller] Randy Miller. *Język UMLTM w praktyce: wprowadzenie dla programistów*. Internet: www.borland.pl/tech/poradnik_uml.shtm.
- [OCL03] *UML 2.0 OCL Specification*. Object Management Group, No. ptc/03-10-14, 2003. Internet: www.omg.org.
- [OMG-FTF] Issues for MOF 2.0/UML 2.0 Infrastructure Finalization Task Force. *Issue 6464: UML 2 Issue: isUnique*. Internet: www.omg.org/issues/mu2i-fff.open.html#Issue6464.
- [Övergaard98] Gunnar Övergaard. A Formal Approach to Relationships in The Unified Modeling Language. *Proceedings PSMT'98 Workshop on Precise Semantics for Modeling Techniques*, Technische Universität München, 1998.
- [Płodzień03] Jacek Płodzień, Ewa Stemposz. *Analiza i projektowanie systemów informatycznych*. Wydawnictwo PJWSTK, Warszawa, 2003.
- [Pons02] Claudia Pons. Generalization Relation in UML Model Elements. *Proceedings of the Inheritance Workshop at ECOOP 2002*, Málaga, Spain, pp. 69–75, 2002.

- [Rumbaugh91] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall, Englewood Cliffs, N.J., 1992.
- [Rumbaugh04] James Rumbaugh, Ivar Jacobson, Grady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 2nd ed., 2004.
- [Rumpe98] Bernhard Rumpe. A Note on Semantics (with an Emphasis on UML). *Proceedings Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications)*, pp. 177–197, 1998.
- [Sacha96] Krzysztof Sacha. *Projektowanie oprogramowania systemów sterujących*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1999.
- [Schmidt86] David A. Schmidt. *Denotational Semantics: A Methodology for Language Development*. Allyn and Bacon, Boston, MA, 1986.
- [Shlaer88] Sally Shlaer, Stephen J. Mellor. *Object-Oriented Systems Analysis: Modeling the World in Data*. Prentice Hall, Englewood Cliffs, N.J., 1988.
- [Spivey92] Michael J. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall, 2nd ed., 1992.
- [Stevens02] Perdita Stevens. On the Interpretation of Binary Associations in the Unified Modelling Language. *Software and Systems Modeling*, Vol. 1, No. 1, pp. 68–79, 2002.
- [Subieta99a] Kazimierz Subieta. *Słownik terminów z zakresu obiektowości*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1999.
- [Subieta99b] Kazimierz Subieta. Język UML. *Materiały V Konferencji PLOUG (Oracle)*, Zakopane, 1999.
- [Szejko02] Stanisław Szejko (Ed.). *Metody wytwarzania oprogramowania*. Wydawnictwo MIKOM, Warszawa, 2002.
- [Szlenk02] Marcin Szlenk. Podstawowe pojęcia modelu UML w ujęciu formalnym. *Materiały IV Krajowej Konferencji Inżynierii Oprogramowania*, Tarnowo Podgórne k. Poznania, pp. 173–180, 2002.
- [Szlenk03] Marcin Szlenk. Formalizacja procesu konstrukcji obiektowych modeli danych. *Materiały X Konferencji Systemy Czasu Rzeczywistego*, Ustroń, pp. 71–80, 2003.

- [Szlenk04] Marcin Szlenk. Precyzyjna semantyka pojęciowych diagramów klas UML. *IV Międzynarodowe Warsztaty Doktoranckie OWD 2004*, Wisła, Vol. 4, pp. 205–210, 2004.
- [Trzęsicki03] Kazimierz Trzęsicki. *Logika i teoria mnogości*. Akademicka Oficyna Wydawnicza Exit, Warszawa, 2003.
- [UML03] *Unified Modeling Language Specification, Version 1.5*. Object Management Group, No. formal/03-03-01, 2003. Internet: www.omg.org.
- [UML03a] *UML 2.0 Infrastructure Specification*. Object Management Group, No. ptc/03-09-15, 2003. Internet: www.omg.org.
- [UML03b] *UML 2.0 Diagram Interchange Specification*. Object Management Group, No. ptc/03-09-01, 2003. Internet: www.omg.org.
- [UML04] *UML 2.0 Superstructure Specification*. Object Management Group, No. ptc/04-10-02 (convenience document), 2004. Internet: www.omg.org.
- [Ward85] Paul T. Ward, Stephen J. Mellor. *Structured Development for Real-Time Systems*. Yourdon Press, Englewood Cliffs, N.J., 1985.
- [Warmer03] Jos Warmer, Anneke Kleppe. *OCL: precyzyjne modelowanie w UML*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2003.
- [Winskel93] Glynn Winskel. *The Formal Semantics of Programming Languages: An Introduction*. The MIT Press, Cambridge, MA, 1993.
- [Yourdon96] Edward Yourdon. *Współczesna analiza strukturalna*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1996.

Dodatek A

Słownik terminów anglojęzycznych

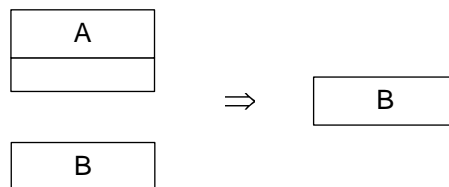
<i>aggregate</i>	agregat
<i>aggregation</i>	agregacja
<i>analysis pattern</i>	wzorzec analizy
<i>association</i>	asocjacja
<i>association class</i>	klasa asocjacji
<i>attribute</i>	atrybut
<i>behavior diagrams</i>	diagramy zachowania
<i>class</i>	klasa
<i>classifier</i>	klasyfikator
<i>conceptual class</i>	klasa pojęciowa
<i>conceptual class diagram</i>	pojęciowy diagram klas
<i>conceptual domain</i>	dziedzina pojęciowa
<i>conceptual model</i>	model pojęciowy
<i>conceptual perspective</i>	perspektywa pojęciowa
<i>constraint</i>	ograniczenie
<i>composite</i>	kompozyt
<i>composition</i>	kompozycja
<i>dependency</i>	zależność
<i>directed association</i>	asocjacja skierowana
<i>domain</i>	dziedzina
<i>domain model</i>	model dziedziny
<i>discriminator</i>	dyskryminator
<i>extensibility mechanism</i>	mechanizm rozszerzalności
<i>extent</i>	ekstensja
<i>generalization</i>	generalizacja
<i>generalization set</i>	zbiór generalizacji
<i>identity</i>	tożsamość
<i>inheritance</i>	dziedziczenie
<i>instance</i>	instancja

<i>implementation perspective</i>	perspektywa implementacyjna
<i>leaf</i>	liść
<i>link</i>	powiązanie
<i>meta-language</i>	metajęzyk
<i>meta-model</i>	metamodel
<i>method</i>	metoda
<i>methodology</i>	metodyka
<i>model</i>	model
<i>multiplicity</i>	liczność
<i>object</i>	obiekt
<i>operation</i>	operacja
<i>problem domain</i>	dziedzina problemu
<i>profile</i>	profil
<i>property</i>	własność
<i>realization</i>	realizacja
<i>responsibility</i>	zobowiązanie
<i>role</i>	rola
<i>semantic domain</i>	dziedzina semantyczna
<i>semantic mapping</i>	odwzorowanie semantyczne
<i>software perspective</i>	perspektywa oprogramowania
<i>specialization</i>	specjalizacja
<i>specification perspective</i>	perspektywa specyfikacyjna
<i>stereotype</i>	stereotyp
<i>structure diagrams</i>	diagramy struktury
<i>subclass</i>	podklasa
<i>superclass</i>	nadklasa
<i>syntactical domain</i>	dziedzina syntaktyczna
<i>tagged value</i>	wartość etykietowana
<i>visibility</i>	widoczność

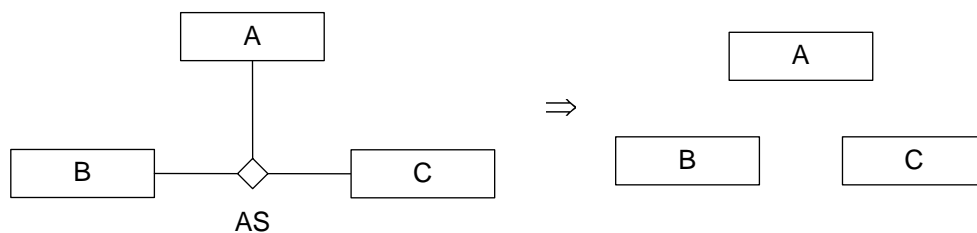
Dodatek B

Zestawienie reguł wynikania

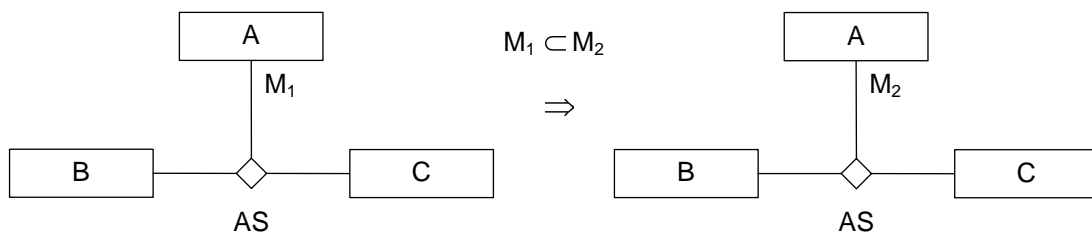
1. Usunięcie klasy (str. 43)



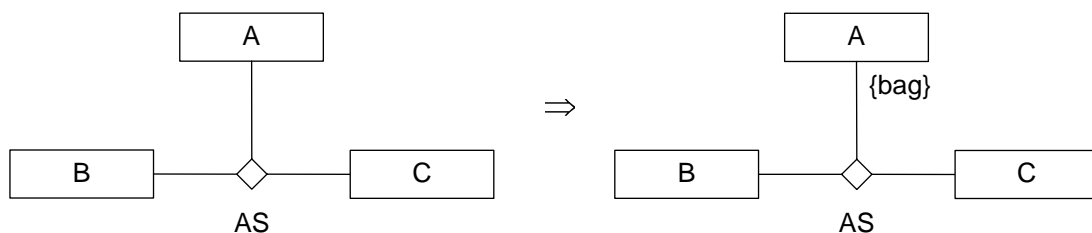
2. Usunięcie asocjacji (str. 44)



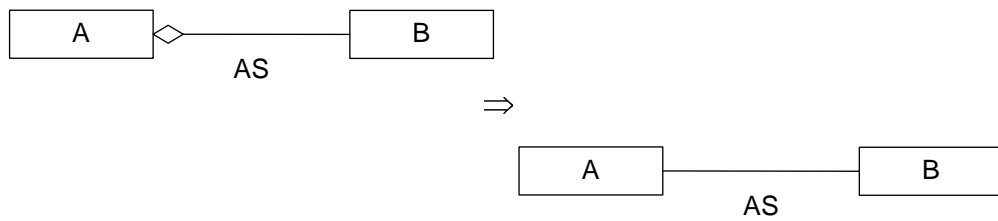
3. Rozszerzenie liczności końca asocjacji (str. 45)



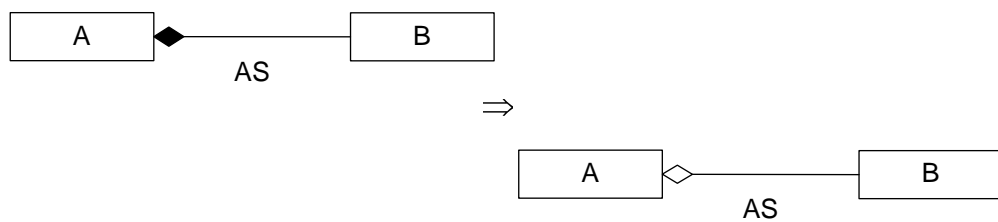
4. Usunięcie unikalności powiązań (str. 46)



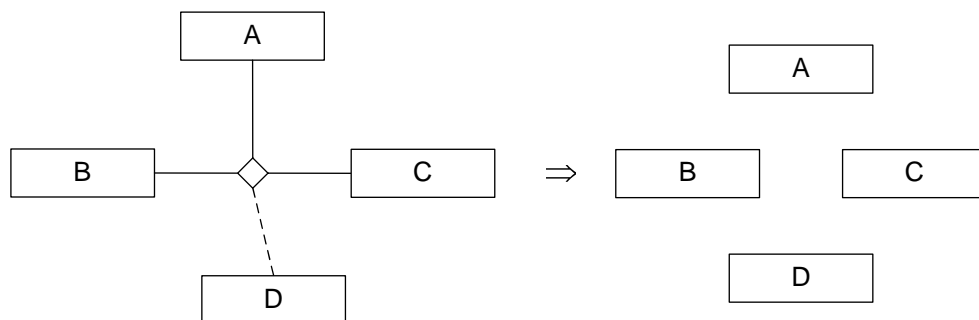
5. Osłabienie agregacji(str. 48)



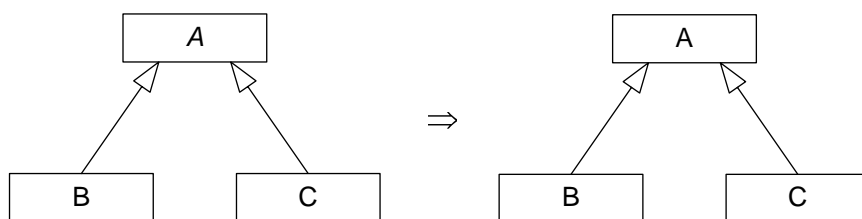
6. Osłabienie kompozycji(str. 50)



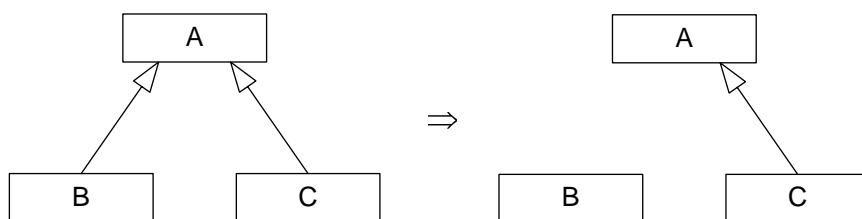
7. Przekształcenie klasy asocjacji w klasę (str. 51)



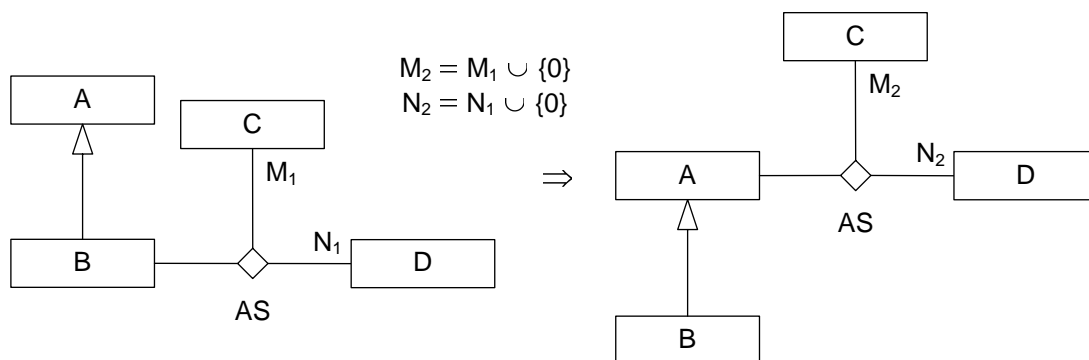
8. Usunięcie abstrakcyjności klasyfikatora (str. 52)



9. Usunięcie związku specjalizacji (str. 53)



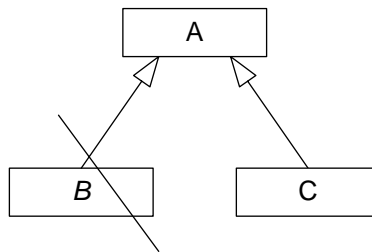
10. Promocja asocjacji (str. 55)



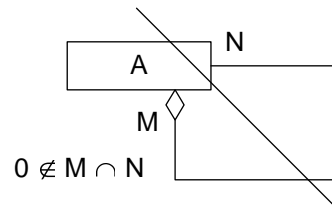
Dodatek C

Zestawienie wzorców niespójności

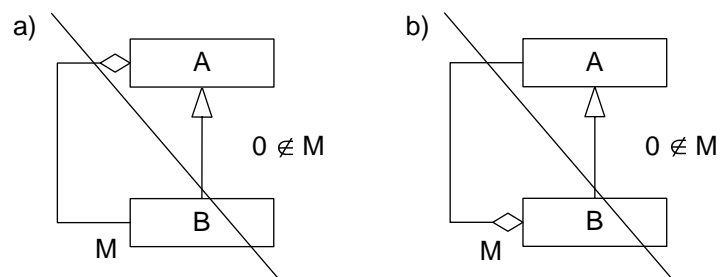
1. Niespójność abstrakcyjnego liścia (str. 37)



2. Niespójność dla agregacji rekursywnej (str. 40)



3. Niespójność dla złożonej agregacji rekursywnej (str. 61)



4. Niespójność dla kompozycji (str. 68)

