Specyfikacje formalne

Dr hab. inż. Ilona Bluemke

Typy specyfikacji formalnych

Specyfikacje formalne można podzielić na dwa typy:

- Algebraiczne
- Bazujące na modelu.

Specyfikacje algebraiczne

Obiekt jest specyfikowany jako zbiór relacji między operacjami na tym obiekcie. Po raz pierwszy specyfikacja algebraiczna wprowadzona była przez Guttag'a w 1977 do specyfikacji abstrakcyjnych typów danych.

W specyfikacjach algebraicznych są notacje **sekwencyjne**:

- **OBJ** (Futatsugi.. 1985),
- **Larch** (Guttag.. 1985),

oraz notacje **współbieżne**: **Lotos** (Bolognesi , Brinksma 1987)

Specyfikacje algebraiczne-2

Specyfikacja algebraiczna określa operacje na obiekcie i specyfikuje je podając relacje między nimi. Składa się z dwóch części:

- **sygnaturowej** - zawierającej operacje i ich parametry,
- **aksjomatycznej** - zawierającej definicje operacji.

Można tworzyć nowe typy dziedziczące operacje i aksjomaty istniejących już typów i dodając nowe operacje i aksjomaty.

Współrzędne kartezjańskie

sort Coord

imports Integer, Boolean;

Specyfikacja def. Coord - reprezentujący współrzędne kartezjańskie. Operacje zdefiniowane dla Coord to X, Y, które obliczają atrybuty oraz Eq porównująca 2 obiekty typu Coord.

Create (Integer, Integer) -> Coord;

X (Coord) -> Integer;

Y (Coord) -> Integer;

Eq (Coord,Coord) -> Boolean;

X (Create(x,y)) = x;

Y (Create(x,y)) = y;

Eq(Create(x1,y1),Create(x2,y2)) = ((x1=x2) and (y1=y2))

Tworzenie specyfikacji algebraicznych

- Strukturalizacja specyfikacji (wybór typów abstrakcyjnych).
- Nazwanie specyfikacji, określenie, czy ma ona parametry generyczne.
- Wybór operacji np. tworzenia, modyfikacji, inspekcji.
- Nieformalna specyfikacja operacji.
- Definicja składni i parametrów.
- Definicja aksjomatów, określenie semantyki operacji poprzez określenie warunków, które są zawsze spełnione dla różnych kombinacji operacji.

Specyfikacji algebraiczna dla **listy**

sort Lista
imports Integer

Specyfikacja definiuje listę, której elementy są dodawane na końcu, a usuwane z początku. Operacja Create tworzy pustą listę, a Cons dodaje element do listy. Operacja Length podaje liczbę elementów listy, Head podaje początkowy element listy a Tail usuwa pierwszy element z listy.

Create -> List
Cons (List,element) -> List
Head (List) -> element
Lenght(List) -> Integer
Tail (List) -> List

Head (Create) = undefined **exception** (empty list)
Head (Cons(L,v)) = **if** L = Create **then** v **else** Head (L)
Lenght (Create) = 0
Lenght (Cons(L,v)) = Lenght (L) + 1
Tail (Create) = Create
Tail (Cons (L,v)) = **if** L = Create **then** Create **else** Cons (Tail (L),v)

Rekursja w specyfikacji algebraicznej

Operacja Tail jest zdefiniowana na pustej liście, a potem rekursywnie na liście niepustej, rekursja kończy się, gdy lista jest pusta.

Przy specyfikacji rekursywnej warto jest sprawdzić ją na prostym przykładzie.

Tail ([3,4,5]) = Tail (Cons ([3,4], 5)) =

Cons (Tail ([3,4], 5)) = Cons (Tail (Cons ([3], 4)), 5) =

Cons (Cons (Tail ([3]), 4), 5) =

Cons (Cons (Tail (Cons ([]), 3)), 4), 5) =

Cons (Cons ([Create], 4), 5) = Cons ([4],5) = [4,5]

Specyfikacja bazująca na modelu

może być stosowana, gdy system modelowany jest takimi obiektami jak zbiory, funkcje.

Istnieją techniki **sekwencyjne** np. :

- **VDM** (Viena Definition Method - Jones 1980, 1986),
- **Z** (Zed) (Abrial 1980, rozwijany w Oxford Hayes 1986, Spivey 1990)

oraz współbieżne.

Z-schematy

Dla Z-schematów są dostępne programy do tworzenia i sprawdzania specyfikacji.

Specyfikacja w Z jest pewną liczbą schematów.

Każdy schemat wprowadza określonego typu nazwę i definiuje predykaty dla tej nazwy.

Schematy mogą być prezentowane w grafice. Są "bloczkami" do budowy innych schematów.

Przykład Z-schematu

pojemnik *nazwa*

---

zawartość : N *sygnatury*

pojemność : N

---

zawartość <= pojemność *predykat schematu*

Z-schematy

- Można "składać" schematy tak by dziedziczyły sygnatury, predykaty.
- Można określać

  operacje **wejścia** nazwa**?**: N

  operacje **wyjścia** nazwa**!**: N
- Jeśli nazwa schematu jest poprzedzona grecką literą Δ (delta schemat) oznacza to, że jedna lub więcej wartości zmiennych stanu **zostanie zmienionych** przez operacje. Dla wszystkich zmiennych wprowadzonych w schemacie można wprowadzić także zmienną**'** określającą zmienioną wartość zmiennej.

Z-schematy-2

- Jeśli nazwa schematu jest poprzedzona grecką literą Ξ (Xi schemat) oznacza to, że wartości zmiennych stanu **nie zostaną zmienione** przez operacje (zmienna**'** nie zmieniona).
- Można korzystać z funkcji, także takich, które nie dla wszystkich dozwolonych wejść (dziedzina) mają określone wartości wyjścia (**partial**). Są operatory pozwalające na działania na dziedzinie.
- Jeśli istotna jest kolejność to można użyć sekwencji (**sequence**).

Schemat wskaźnik

wskaźnik

---

lampka : {on, off}

odczyt : N

niebezpieczny_poziom : N

---

lampka = on $\Leftrightarrow$ odczyt $\leq$ niebezpieczny_poziom

Schemat zbiornik

Zmienne stanu schematów (odczyt , zawartość) użyte są do połączenia obu schematów.

zbiornik

---

pojemnik

wskaźnik

---

odczyt = zawartość

pojemność = 4000

niebezpieczny_poziom = 40

Schemat **napełnianie_OK**

napełnianie_OK

Δ zbiornik

ilość? : N

---

zawartość + ilość? $\leq$ pojemność

zawartość’ = zawartość + ilość

Schemat przepełnienie

- oznacza, że zmienne stanu tego schematu **nie mogą zostać zmienione**, jest użyty w schemacie

przepełnienie

- zbiornik

ilość? : N

t! : seq char

---

pojemność < zawartość + ilość?

t! = „niewystarczająca pojemność zbiornika – napełnianie skasowane”

Schemat napełnianie

Następnie schematy **przepełnienie** oraz **napełnianie_OK** są połączone w definicji schematu **napełnianie.**

napełnianie

---

napełnianie_OK $\cup$ przepełnienie