

3.2. Abstrakcyjne typy danych

Przyklad 3.4

Kolejka: Queue

a) opis nieformalny operacji kolejki

EMPTY:	Inicjowanie pustej kolejki
ADD:	Dodanie nowego elementu na koncu kolejki
FRONT:	Wskazanie elementu na początku kolejki
REMOVE:	Usunięcie elementu z początku kolejki
IS_EMPTY?:	Sprawdzenie, czy kolejka jest pusta

b) syntaktyczny opis operacji kolejki

EMPTY:		--->	Queue
ADD:	Queue x Item	--->	Queue
FRONT:	Queue	--->	Item
REMOVE:	Queue	--->	Queue
IS_EMPTY?:	Queue	--->	Boolean

c) aksjomatyczny opis operacji kolejki

- (1) $IS_EMPTY?(EMPTY) = true$
- (2) $IS_EMPTY?(ADD(q, i)) = false$
- (3) $FRONT(EMPTY) = error$
- (4) $FRONT(ADD(q,i)) = \mathbf{if\ IS_EMPTY?(q)\ then\ i}$
 $\qquad\qquad\qquad \mathbf{else\ FRONT(q)}$
- (5) $REMOVE(EMPTY) = error$
- (6) $REMOVE(ADD(q,i)) = \mathbf{if\ IS_EMPTY?(q)\ then\ EMPTY}$
 $\qquad\qquad\qquad \mathbf{else\ ADD(REMOVE(q),i)}$

Uwaga:

Wszelkie operacje na kolejce niezainicjowanej dają w wyniku błąd, czyli: ADD, FRONT, REMOVE, IS_EMPTY?, gdyż operacje na argumentach błędnych są błędne.

Dowód niesprzeczności:

Ponieważ znaczenie operacji jest określone przez zbiór zdań (aksjomatów), specyfikacja jest sprzeczna, jeżeli jakiegokolwiek dwa z tych zdań są sprzeczne

Na podstawie (5) i (1) mamy:

$$\begin{aligned} \text{IS_EMPTY?}(\text{REMOVE}(\text{EMPTY})) &= \text{IS_EMPTY?}(\text{error}) \\ &= \text{error} \end{aligned}$$

Na podstawie (5) i (2) mamy:

$$\begin{aligned} \text{IS_EMPTY?}(\text{ADD}(\text{REMOVE}(\text{EMPTY}),i)) &= \text{IS_EMPTY?}(\text{ADD}(\text{error},i)) \\ &= \text{IS_EMPTY?}(\text{error}) \\ &= \text{error} \end{aligned}$$

Na podstawie (6) i (1) mamy:

$$\begin{aligned} \text{IS_EMPTY?}(\text{REMOVE}(\text{ADD}(\text{EMPTY},i))) &= \text{IS_EMPTY?}(\text{EMPTY}) \\ &= \text{true} \end{aligned}$$

Na podstawie (6) i (2) mamy:

$$\begin{aligned} \text{IS_EMPTY?}(\text{REMOVE}(\text{ADD}(q,i))) &= \text{IS_EMPTY?}(\text{ADD}(\text{REMOVE}(q),i)) \\ &= \text{IS_EMPTY?}(\text{ADD}(\text{EMPTY},i)) \\ &= \text{false} \end{aligned}$$

Na podstawie (3) i (5) mamy:

$$\begin{aligned} \text{FRONT}(\text{REMOVE}(\text{EMPTY})) &= \text{FRONT}(\text{error}) \\ &= \text{error} \end{aligned}$$

Na podstawie (3) i (6) mamy:

$$\begin{aligned} \text{FRONT}(\text{REMOVE}(\text{ADD}(\text{EMPTY},i))) &= \text{FRONT}(\text{EMPTY}) \\ &= \text{error} \end{aligned}$$

Na podstawie (4) i (6) mamy:

$$\begin{aligned} \text{FRONT}(\text{REMOVE}(\text{ADD}(q,i))) &= \text{FRONT}(\text{ADD}(\text{REMOVE}(q),i)) \\ &= \text{FRONT}(\text{ADD}(\text{EMPTY},i)) \\ &= i \end{aligned}$$

oraz

$$\begin{aligned} \text{FRONT}(\text{REMOVE}(\text{ADD}(q,i))) &= \text{FRONT}(\text{ADD}(\text{REMOVE}(q),i)) \\ &= \text{FRONT}(\text{ADD}(q1,i)) \\ &= \text{FRONT}(q1) \end{aligned}$$

Opis jest niesprzeczny.

Przykład 3.5

Vector

a) opis nieformalny operacji na wektorze elementów

EMPTY:	Zainicjowanie wektora pustego
ASSIGN:	Przypisanie do zainicjowanego wektora na wybranej pozycji elementu
READ:	Pobranie z zainicjowanego wektora elementu z wybranej pozycji
IS_EMPTY?:	Sprawdzenie, czy zainicjowany wektor jest pusty

b) opis syntaktyczny operacji na wektorze elementów

EMPTY:		--->	Vector
ASSIGN:	Vector x Index x Item	--->	Vector
READ:	Vector x Index	--->	Item
IS_EMPTY?:	Vector	--->	Boolean

c) opis aksjomatyczny operacji wektorze elementów

- (1) IS_EMPTY?(EMPTY) = true
- (2) IS_EMPTY?(ASSIGN(v,i,a)) = false
- (3) ASSIGN(v,i,a) = v
- (4) READ(ASSIGN(v,i1,a),i2) = **if** IS_EQUAL? (i1,i2) **then** a **else** READ(v,i2)

Dowód niesprzeczności:

Na podstawie (2) mamy

IS_EMPTY?(ASSIGN(EMPTY,i,a)) = false

Na podstawie (1) i (3)

IS_EMPTY?(ASIGN(EMPTY,i,a)) = IS_EMPTY?(EMPTY)
= true

czyli false=true, czyli wystąpiła sprzeczność.

Dowód zupełności:

Brak określenia wartości READ(EMPTY,i), czyli:

READ(EMPTY,i) = error