



Datenstrukturen und Algorithmen

Tutorium 1, KW 16 2013

Aufgabe T1.1: Abstrakter Datentyp I

Punkte in der euklidischen Ebene stellen bzgl. eines gegebenen Koordinatensystems folgende Operationen zur Verfügung:

create	liefert zu zwei Zahlen einen Punkt, dessen x-Koordinate gleich der ersten Zahl ist und dessen y-Koordinate gleich der zweiten Zahl ist.
get _x	liefert die x-Koordinate eines Punktes.
get _y	liefert die y-Koordinate eines Punktes.
isOrigin	prüft, ob ein Punkt der Ursprung (0, 0) ist.
translate	verschiebt einen Punkt um eine erste Zahl in x-Richtung und um eine zweite Zahl in y-Richtung.
scale	skaliert die Koordinaten eines Punktes um eine Zahl.
distance	liefert den Abstand zweier Punkte.

Geben Sie Sorten, Funktionen und Axiome an, welche einen abstrakten Datentyp (Abk.: ADT) für Punkte definieren. Die Grundrechenarten können dabei als elementar betrachtet werden.

Aufgabe T1.2: Abstrakter Datentyp II

Gegeben sei folgende Definition eines ADTs:

Sorten: CALC, INTEGER

Funktionen:

create	:		→	CALC
put	:	CALC × INTEGER	→	CALC
add	:	CALC	→	CALC
length	:	CALC	→	INTEGER
get	:	CALC	→	INTEGER

Axiome: $\forall c \in \text{CALC} \ \forall i, j \in \text{INTEGER}$

- (1) $\text{add}(\text{create}()) = \text{error}$
- (2) $\text{add}(\text{put}(\text{create}(), i)) = \text{error}$
- (3) $\text{add}(\text{put}(\text{put}(c, i), j)) = \text{put}(c, i + j)$
- (4) $\text{length}(\text{create}()) = 0$
- (5) $\text{length}(\text{put}(c, i)) = \text{length}(c) + 1$
- (6) $\text{get}(\text{create}()) = \text{error}$
- (7) $\text{get}(\text{put}(c, i)) = i$

Die Grundrechenarten werden dabei als elementar betrachtet.

1. Begründen Sie, dass mit den Axiomen (1)-(7) alle validen Terme über $\{\text{create}, \text{put}, \text{add}, \text{length}, \text{get}\}$ in einen Term der Form $\text{put}(\text{put}(\dots, n_2), n_1)$, in $\text{create}()$ oder in eine INTEGER-Zahl überführt werden können. Ein Term ist valide, wenn er nicht in einen error überführt werden kann.
2. Vereinfachen Sie die folgenden Terme unter Angabe des in jedem Schritt verwendeten Axioms.
 - (a) $\text{get}(\text{put}(\text{add}(\text{put}(\text{put}(\text{create}(), 1), 3)), 5))$
 - (b) $\text{length}(\text{put}(\text{add}(\text{put}(\text{put}(\text{create}(), 1), 3)), 5))$
3. Beschreiben Sie den Datentyp mit eigenen Worten.
4. Beweisen Sie folgende Behauptung:

$$\forall x, y \in \text{INTEGER} \quad \text{add}(\text{put}(\text{put}(\text{create}(), x), y)) = \text{add}(\text{put}(\text{put}(\text{create}(), y), x))$$
5. Spezifizieren Sie eine Operation sum (Signatur und Axiome), die alle in CALC gespeicherten Werte aufsummiert und analog zu add alle bearbeiteten Werte durch das Ergebnis ersetzt.

Aufgabe T1.3: Terme

Gegeben seien die Terme

$$t_1 = (2 + 3) * 4 + 15 / (12 - 4 - 3)$$

$$t_2 = 2 * 3 * 4 + 8 * (4 - 3) / 2$$

1. Bestimmen Sie jeweils eine Postfix-Darstellung zu t_1 und t_2 .
2. Werten Sie die Postfix-Darstellung von t_1 mit einem Stack-Computer aus. Geben Sie den Inhalt des Stacks nach jeder Push- oder Pop-Operation an.