SETS (MENGEN)

Formal:

Sei M eine geg. Menge von Datenelementen;

- Menge aller endlichen Teilmengen $A \subseteq M$ einer Grundmenge M;
- $A = \{a_1, a_2, \ldots, a_n\}$ oder $A = \{a \in M | P(a)\}$,
- leere Menge $\emptyset$

als ADT:

SET(M) besteht aus:

- der Menge aller endlichen Teilemengen $A \subseteq M$ einer Grundmenge M;
- den folgenden Operationen (bzw. einer Auswahl daraus):

Operationen des ADT Set(M)

(Sei A: Set(M) und Elem eine Implementierung der Grundmenge M)

clear(): Macht A zur leeren Menge

(V) -

(N) $A' = \emptyset$

isEmpty(): Test auf leere Menge

(V) -

(N) A.isEmpty = TRUE, falls $A = \emptyset$, else FALSE

boolean member(Elem x): Elementrelation

(V) -

(N) A.member(x) = TRUE, falls $x \in A$
A.member(x) = FALSE, falls $x \notin A$

insert(Elem x): Hinzufügen von x zu A, d. h. $A := A \cup \{x\}$

(V) -

(N) $x \in A'$

delete(Elem x): Entfernen von x aus A, d. h. $A := A - \{x\}$

(V) -

(N) $x \notin A'$

union(Set B): Vereinigung

(V) -

(N) $A' = A \cup B$

intersection(Set B): Durchschnitt

(V) -

(N) $A' = A \cap B$

difference(Set B): Mengendifferenz

(V) -

(N) $A' = A - B$

boolean equals(Set B): Test auf Gleichheit

(V) -

(N) A.equals(B) = TRUE, falls $A = B$
A.equals(B) = FALSE, falls $A \neq B$

boolean subset(Set B): Test auf Enthaltensein

(V) -

(N) A.subset(B) = TRUE, falls $A \subset B$
A.subset(B) = FALSE, falls $A \not\subset B$

assign(Set B): Zuweisung

(V) -

(N) $A' = B$

int card(): Kardinalität (Anzahl der Elemente in A)

(V) $A = \{a_1, a_2, \ldots, a_n\}$, $n \geq 0$

(N) A.card = n

Weitere Operationen:

Wir setzen voraus, daß auf der Grundmenge *M* eine totale (lineare) Ordnung $\leq \ \subseteq M \times M$ existiert, d. h. $\leq$ ist

1. reflexiv: $\forall a \in M(a \leq a)$,
2. antisymmetrisch: $\forall a, b \in M(a \leq b \wedge b \leq a \Rightarrow a = b)$,
3. transitiv: $\forall a, b, c \in M(a \leq b \wedge b \leq c \Rightarrow a \leq c)$,
4. total: $\forall a, b \in M(a \leq b \vee b \leq a)$.

Weitere denkbare Operationen auf SET(*M*) sind dann

Elem min():

(V) $A \neq \emptyset$

(N) A.min = dasjenige $a \in A$ mit $\forall b \in A(a \leq b)$,
undefiniert, falls $A = \emptyset$

Elem max():

(V) $A \neq \emptyset$

(N) A.max = dasjenige $a \in A$ mit $\forall b \in A(b \leq a)$,
undefiniert, falls $A = \emptyset$

Und die bekannten vier Operationen, die es ermöglichen, Mengen von Mengen der Form Set(Set(*M*)) zu bilden (vgl. ADT List(*A*)).

Set copy():	Kopieroperation
boolean equals (Set b):	Test auf Gleichheit
boolean less(Set b):	Test auf “Kleiner”
StringtoString():	Konvertierungsoperation

Informatik 1, Sets (Mengen) WS 2005

MENGENIMPLEMENTIERUNG ÜBER BITVEKTOR

Im folgenden sei gegeben:

SetT A = new BitSetT();

Operationen:

A durch $\bar{b}_A$ dargestellt, $\bar{b}_A(i) \;\widehat{=}\; a_i$

A.clear() : $\forall i : \bar{b}_A(i) := 0$

A.isEmpty() : $\forall i : \bar{b}_A(i) = 0$?

A.member (a_i) : $\bar{b}_A(i) = 1$? 0(1)

A.insert (a_i) : $\bar{b}_A(i) := 1$ 0(1)

A.delete (a_i) : $\bar{b}_A(i) := 0$ 0(1)

A.union (B) : $\forall i$: $\bar{b}_A(i) := \bar{b}_A(i) \vee \bar{b}_B(i)$

A.intersection (B): $\forall i$: $\bar{b}_A(i) := \bar{b}_A(i) \wedge \bar{b}_B(i)$

A.difference (B) : $\forall i$: $\bar{b}_A(i) := \bar{b}_A(i) \wedge\neg\, \bar{b}_B(i)$

A.equals (B) : $\forall i$: $\bar{b}_A(i) = \bar{b}_B(i)$?

A.subset (B) : $\forall i$: $\bar{b}_A(i) \rightarrow \bar{b}_B(i)$?

A.assign (B) : $\forall i$: $\bar{b}_A(i) := \bar{b}_B(i)$

A.card() : $\sum_{i=1}^{n} \bar{b}_A(i)$

Es gelte: $a_1 < a_2 < \ldots a_n$

A.min: a_i, wobei i = min {k | $\bar{b}_A(k) = 1$}

A.max: a_i, wobei i = max {k | $\bar{b}_A(k) = 1$}

ZEITKOMPLEXITÄT DER OPERATIONEN:

<table>
<tr><th>Operation</th><th>Bitvektor</th><th>Zeiger (2.3)</th></tr>
<tr><td>clear</td><td>O (n)[1)]</td><td>O (1)</td></tr>
<tr><td>isEmpty</td><td>O (n)[1)]</td><td>O (1)</td></tr>
<tr><td>member</td><td>O (1)</td><td>O (n) → locate</td></tr>
<tr><td>insert</td><td>O (1)</td><td>O (n) → locate; insert;</td></tr>
<tr><td>delete</td><td>O (1)</td><td>O (n) → locate; delete</td></tr>
<tr><td>union</td><td rowspan="7">O (n)[1)]</td><td rowspan="6">O (n^2), falls unsortiert
O (n+n) = O (2n) = O (n), falls sortiert</td></tr>
<tr><td>intersection</td></tr>
<tr><td>difference</td></tr>
<tr><td>equals</td></tr>
<tr><td>subset</td></tr>
<tr><td>assign</td></tr>
<tr><td>assign</td><td>O (1)</td></tr>
<tr><td>card</td><td>O (1) / O (n)[2)]</td><td>O (1) / O (n)[2)]</td></tr>
<tr><td>min</td><td rowspan="2">O (n)</td><td rowspan="2">O (n), falls unsortiert
O (1), falls sortiert</td></tr>
<tr><td>max</td></tr>
</table>

1) u. U. effektiv durch Maschinenoperationen umsetzbar, falls Bitvektor = Maschinenwort;

2) falls Instanzvariable für Kardinalität nicht geführt wird;