

1 Datenstrukturen

- 1.1 Abstrakte Datentypen
- 1.2 Lineare Strukturen
- 1.3 Bäume
- 1.4 Prioritätsschlangen
- 1.5 Graphen



Abstrakte Datentypen

- Spezifizieren Form und Funktionalität der zu verarbeitenden Daten
- Datenobjekte, Datenfelder, “Sorten”
- Funktionen
- Axiome



Abstrakte Datentypen

- Spezifizieren Form und Funktionalität der zu verarbeitenden Daten
- Datenobjekte, Datenfelder, “Sorten”
- Funktionen
- Axiome

Zeit: hh:mm:ss



Abstrakte Datentypen

- Spezifizieren Form und Funktionalität der zu verarbeitenden Daten
- Datenobjekte, Datenfelder
- Funktionen
- Axiome

Add: $\text{Zeit} \times \text{Zeit} \rightarrow \text{Zeit}$



Abstrakte Datentypen

- Spezifizieren Form und Funktionalität der zu verarbeitenden Daten
- Datenobjekte, Datenfelder
- Funktionen
- **Axiome**

$$\begin{aligned} \text{Add}([h_1, m_1, s_1], [h_2, m_2, s_2]) = \\ [(h_1 + h_2 + (m_1 + m_2 + (s_1 + s_2) / 60) / 60) \% 24, \\ (m_1 + m_2 + (s_1 + s_2) / 60) \% 60, \\ (s_1 + s_2) \% 60] \end{aligned}$$



Abstrakte Datentypen

- Warum abstrakt?
 - Keine Hinweise darauf, wie die jeweiligen Funktionen implementiert werden.
- Warum Typdefinitionen?
 - Spezifikation / Verifikation
 - Top-down Software-Entwurf



Beispiel: Bool

- Wertebereich: { true, false }
- $\neg: \text{bool} \rightarrow \text{bool}$
 $\wedge: \text{bool} \times \text{bool} \rightarrow \text{bool}$
 $\vee: \text{bool} \times \text{bool} \rightarrow \text{bool}$
- $\neg \text{true} = \text{false}$, $\neg \text{false} = \text{true}$
 $x \wedge \text{true} = x$, $x \wedge \text{false} = \text{false}$,
 $x \vee \text{true} = \text{true}$, $x \vee \text{false} = x$



Beispiel: Bool

- Wertebereich: { true, false }
- \neg : bool \rightarrow bool
- \wedge : bool \times bool \rightarrow bool
- \vee : bool \times bool \rightarrow bool
- \neg true = false, \neg false = true

\wedge	true	false
true	true	false
false	false	false

\vee	true	false
true	true	true
false	true	false

Aufzählungstypen

- Endlicher Wertebereich
 - Vollständige Spezifikation durch Tabellen
- Verallgemeinerung
 - Aufzählungstypen: enum
 - z.B. Wochentage, chemische Elemente, ...
 - Kodierung als ganze Zahlen modulo n
(implizite Ordnungsrelation)



Beispiel: Integer

- Wertebereich: $\mathbb{Z} = \mathbb{N}_+ \cup \{0\} \cup -\mathbb{N}_+$
- $+: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$
- $-: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$
- $-: \mathbb{Z} \rightarrow \mathbb{Z}$
- $\times: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$
- $\div: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$
- ...

Beispiel: Integer

$$+(0, b) = b$$

$$+(a+1, b) = +(a, b) + 1$$

$$+(a-1, b) = +(a, b) - 1$$

$$-(b, b) = 0$$

$$-(a+1, b) = -(a, b) + 1$$

$$-(a-1, b) = -(a, b) - 1$$

$$-(b) = -(0, b)$$

$$\times(0, b) = 0$$

$$\times(a+1, b) = +(\times(a, b), b)$$

$$\times(a-1, b) = -(\times(a, b), b)$$

$$\div(a, 0) = \text{ERROR}$$

$$\div(a, b) = 0 \text{ if } |a| < |b|$$

$$\div(a+b, b) = \div(a, b) + 1$$

$$\div(a-b, b) = \div(a, b) - 1$$

Beispiel: Integer

count down to 0

$$+(0, b) = b$$

$$+(a+1, b) = +(a, b) + 1$$

$$+(a-1, b) = +(a, b) - 1$$

$$-(b, b) = 0$$

$$-(a+1, b) = -(a, b) + 1$$

$$-(a-1, b) = -(a, b) - 1$$

$$-(b) = -(0, b)$$

$$\times(0, b) = 0$$

$$\times(a+1, b) = +(\times(a, b), b)$$

$$\times(a-1, b) = -(\times(a, b), b)$$

count up to 0

$$\div(a, b) = 0 \vee |a| < |b|$$

$$\div(a+b, b) = \div(a, b) + 1$$

$$\div(a-b, b) = \div(a, b) - 1$$



Beispiel: Integer

$$+(0, b) = b$$

$$+(a+1, b) = +(a, b) + 1$$

$$+(a-1, b) = +(a, b) - 1$$

$$-(b, b) = 0$$

$$-(a+1, b) = -(a, b) + 1$$

$$-(a-1, b) = -(a, b) - 1$$

$$-(b) = -(0, b)$$

$$\times(0, b) = 0$$

$$\times(a+1, b) = +(\times(a, b), b)$$

$$\times(a-1, b) = -(\times(a, b), b)$$

count down to b

$$\div(a, 0) = \text{ERROR}$$

$$\div(a, b) = 0 \text{ if } |a| < |b|$$

$$\div(a+b, b) = \div(a, b) + 1$$

$$\div(a-b, b) = \div(a, b) - 1$$

count up to b



Beispiel: Integer

$$+(0, b) = b$$

$$+(a+1, b) = +(a, b) + 1$$

$$+(a-1, b) = +(a, b) - 1$$

$$-(b, b) = 0$$

$$-(a+1, b) = -(a, b) + 1$$

$$-(a-1, b) = -(a, b) - 1$$

$$-(b) = -(0, b)$$

$$\times(0, b) = 0$$

$$\times(a+1, b) = +(\times(a, b), b)$$

$$\times(a-1, b) = -(\times(a, b), b)$$

$$\div(a, 0) = \text{ERROR}$$

$$\div(a, b) = 0 \text{ if } |a| < |b|$$

$$\div(a+b, b) = \div(a, b) + 1$$

$$\div(a-b, b) = \div(a, b) - 1$$

Beispiel: Integer

- noch abstrakter ...
0
 $N = \{ S(0), S(S(0)), \dots \}$
 $-N = \{ P(0), P(P(0)), \dots \}$
- $+(0, y) = y$
 $+(S(x), y) = S(+ (x, y))$
 $+(P(x), y) = P(+ (x, y))$
- ...

Skalare Typen

- „Zahlen“ mit 1-dim Wertebereich
 - Char
 - Integer
 - Float, Double
- Achtung: In *realen* Implementierungen haben skalare Typen meistens einen endlichen Wertebereich
 - Integer : $[1-2^{b-1} .. 2^{b-1}]$
 - Double : $[10^{-300} .. 10^{300}]$

Overflow
Error

Zusammengesetzte Typen

- Endliche / feste Kombination von skalaren Typen oder zusammengesetzten Typen
 - k-dim Vektoren
 - Adressen-Eintrag
 - ...
- Mehrdimensionaler Wertebereich



Beispiel: Sequenz

- Wertebereich: $W = \{ \} \cup N \cup N^2 \cup N^3 \cup \dots$
 $\neq 2^N$!!!
- Create : $\rightarrow W$
- Append : $N \times W \rightarrow W$
- Remove: $N \times W \rightarrow W$



Beispiel: Sequenz

- $\text{Append}(x, \{y_1, \dots, y_n\}) = \{y_1, \dots, y_n, x\}$

$\text{Append}(z, \text{Append}(y, \text{Append}(x, \text{Create()}))) = \{x, y, z\}$

$\text{Remove}(\text{Create}()) = \text{Create}()$

$\text{Remove}(x, \text{Append}(x, z)) = z$

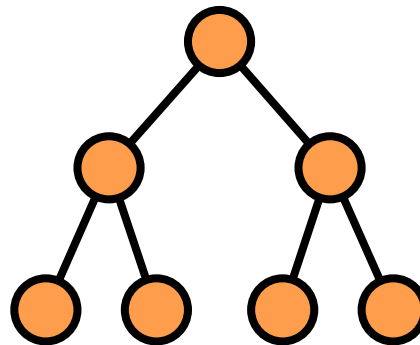
$\text{Remove}(x, \text{Append}(y, z)) =$

$\text{Append}(y, \text{Remove}(x, z))$ if $x \neq y$

- Beliebige Sequenz von Basisobjekten (skalare / zusammengesetzte Typen) mit variabler Länge
 - Arrays
 - Listen (einfach/doppelt verkettet)
 - Queue (FIFO)
 - Stack (LIFO)
- Unterscheiden sich im wesentlichen in der Zugriffsfunktionalität

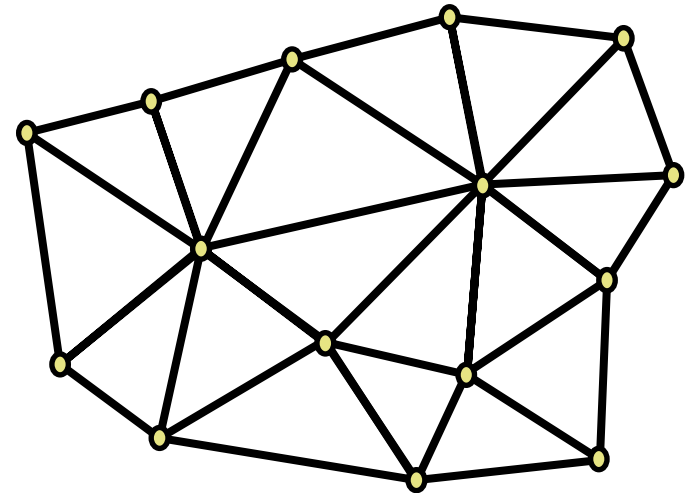
Baumstrukturen

- Hierarchische Strukturen
- Vater/Sohn-Beziehung („der“ Knoten)
- Keine Zyklen
- Eindeutige Vorfahren



Graphen

- Beliebige topologische Struktur
- Nachbarschaftsbeziehungen zwischen Knoten
- Wichtige Spezialfälle
 - Planare Graphen
 - Gerichtete Graphen
 - Zyklentreie Graphen
 - ...



Datentypen-Typen

- Aufzählungstypen (bool, enum)
- Skalare Typen (char, int, float, ..)
- Zusammengesetzte Typen (struct, class)
- Lineare Strukturen (list, queue, stack)
- Bäume
- Graphen



Abstrakte Datentypen

- Spezifizieren Form und Funktionalität der zu verarbeitenden Daten
- Datenobjekte, Datenfelder
- Funktionen
- Axiome



Konkrete Datentypen

- Spezifizieren Form und Funktionalität der zu verarbeitenden Daten
- Datenobjekte, Datenfelder
- Funktionen
- Axiome Methoden / Algorithmen



- Datenobjekte → Attribute
- Funktionen → Methoden-Header (Rückgabetyp, formale Parameter)
- Algorithmen → Methoden-Body
- Überprüfe korrekte Methodenimplementierung anhand der abstrakten Axiome