Algorithm Engineering

April 20, 2015

1 Datentypen

Getränkeautomat

Automat akzeptiert 1E, ein Getränk kostet 3E
Operatoren:

1. Init(Reset)
2. Akzeptiere1E

Init → Zustand
Semantik: Automat geht in Zustand 0
Akzeptiere1E: ZustandX{0,1} → ZustandX{tue nichts, gib Getänk}
Semantik: Beschreibung durch einen endlichen Automaten.

Stadtplan

Übung 1

1.1 Bemerkungen

- Operatoren können partiell Definiert sein. Man gibt Definitionsbereich oft in einer Vorbedingung an.
- Operatoren, bei denen der Datentyp selbst auf der linken Seite nicht vorkommt, heißen Konstruktoren. Sie erzeugen ein neues Objekt (bzw. versetzen den Typ in einem bestimmten Zustand).
  - Create: → stack<T>
  - Create: int → vector (Vektor bestimmter Dimension)
- Objekt- und Zustandssicht sind beide nützlich. Stack/Getränkeautomat haben internen Zustand, Operatoren können ihn verändern.
  - Integer: Objektsicht besser, Operatoren erzeugen neue Objekte, exisitierende werden nicht geändert.
- stack$< T >$ ist ein parametrisierbarer Datentyp: Stack mit Elementen vom Typ T. Hat eventuell besondere Anforederungen an Typ T, z.B. $x \leq y$ in Dictionaries.
- Man kann nun eigentlich schon programmieren, obwohl über die Interpretiernug noch nichts bekannt ist.

Anwendung von stack$< T >$

Auswertung von Postfix-Ausdrücken
Vereinfachungen: alle Operatoren binär (+-*/), Eingabe nur Zahlen 0-9
Bsp.: $(7-5)*(3+1) \rightarrow 75-31+*$

1.2 Defintion eines Datentyps

(In einer Objekt-Orientierten Programmiersprache)

```
class Typname {
        //Definition der Menge der Objekte bzw Zustaende
        private: //Deklaration von Variablen zur Darstellung der Objekte/Zustaende
        public: //Operatoren
        //Kommentare z.B. ueber Effizienz
};
```

Operatoren

Methoden/Memberfunktionen
Syntax: Ergebnistyp Name(Argumente...);
Spezielle Methoden:

- Kein Ergebnistyp: stack(); stack(size);
- Destruktor: $\sim$ Typname();

1.3 Beispiel

int_stack $\rightarrow$ stack$< T >$

```
class int_stack {
        /* Eine Instanz vom Typ int_stack ist eine Folge von ganzen Zahlen (int). Eine Fol
        private: //Implementierung
        public: stack(int sz); //Konstruktor
        //Erzeugt einen Stack mit maximaler Groesse sz
        ~stack() //Destruktor
        void push (int x);
        //fuegt x als letzes Element (top) an die Folge an.
        int top() const;
        //liefert das letzte (top) Element
        //Precondition: Stack nicht leer
        int pop();
        //entfernt letztes (top) Element der Folge und gibt es zurueck
        //Precondition: Stack nicht leer
        bool empty() const;
        //true, wenn Stack leer, false sonst.
```

In c++ Spezielle Header Datei, die die Deklarationen ohne Rumpf enthält. Implementierung in .cpp

Implementierung der Klasse int_stack

Mehrere Möglichkeiten: Array, Liste Array Implementierung:

```
class int_stack {
        private
                int* A; //Feld
                int sz; //Laenge von A
                int t;
};
```