Prof. Dr. Uwe Schmidt

ptlwedel
PHYSIKALISCH-TECHNISCHE LEHRANSTALT

20. August 2004

Aufgaben zur Klausur **C** im SS 2004 (IA 302)

Zeit: 75 Minuten erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 7 Seiten

---

Aufgabe 1:

Gegeben sei der folgende Ausschnitt eines C–Moduls für die Implementierung von binären Suchbäumen.

```
typedef long int Element;

#define compare(x,y) (((x) > (y)) - ((x) < (y)))

typedef struct Node *Set;

struct Node
{
  Element info;
  Set l;
  Set r;
};

Set mkEmptySet(void);
int isEmptySet(Set s);

Set mkOneElemSet(Element e);
Set insertElem(Element e, Set s);
Set removeElem(Element e, Set s);


Set
insertElem(Element e, Set s)
{
  if (isEmptySet(s))
    return mkOneElemSet(e);

  switch (compare(e, s->info))
    {
    case -1:
      s->l = insertElem(e, s->l);
      break;
    case 0:
      break;
    case +1:
      s->r = insertElem(e, s->r);
      break;
    }

  return s;
}
```

```
static
Set removeRoot(Set s) {
  ...
}

Set
removeElem(Element e, Set s)
{
  if (isEmptySet(s))
    return s;

  switch (compare(e, s->info))
    {
    case -1:
      s->l = removeElem(e, s->l);
      break;
    case 0:
      s = removeRoot(s);
      break;
    case +1:
      s->r = removeElem(e, s->r);
      break;
    }

  return s;
}


Set changeElem(Element e, Set s, ...) {
  ...
}

Set
removeElem1(Element e, Set s) {
  return
    changeElem(e, s, ...);
}

Set
insertElem1(Element e, Set s) {
  return
    changeElem(e, s, ...);
}
```

Man erkennt, dass die Funktionen für das Einfügen und Löschen sehr ähnlich sind. Diese Funktionen kann man zusammenfassen zu einer Funktion *changeElem*, die die Unterschiede der beiden Funktionen durch zusätzliche Parameter geliefert bekommt.

Entwickeln Sie die Funktion *changeElem*.
Definieren sie zuerst für die Typen der zusätzlichen Parameter Typdefinitionen:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Der Programmcode für die Funktion *changeElem*.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Der Programmcode für die neue Funktion zum Einfügen *insertElem1* und erforderliche Hilfkonstrukte:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Der Programmcode für die neue Funktion zum Einfügen *removeElem1* und erforderliche Hilfkonstrukte:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Aufgabe 2:

Gegeben seien die folgenden Variablen und Funktionen:

```
typedef int (*Fct1)(int);
typedef int (*Fct2)(int, int);
typedef int (*Fct3)(double);
int x;
long int s;
float f;
char *p1;
int *p2;
void *p3;
Fct1 fp1, fp2;
Fct3 fp3;
int (*pf)(int);
int (*pf2)(double);
int f1(int x1);
int f2(int x1,int x2);
int f3(double x1);
```

Bestimmen Sie für die folgenden Ausdrücke den Typ gemäß ANSI-C. Vorsicht: Es kommen fehlerhafte Ausdrücke von. Kennzeichnen Sie diese entsprechend

p2[x] .....

p3[x] .....

p1 ? x : f .....

∼p2 .....

p1 && p1 .....

p1 & p1 .....

∼s .....

s || x .....

fp1(25) .....

pf=f1 .....

pf=f1(x) .....

pf2=pf .....

fp1==f1 .....

fp1==fp2 .....

f=f3(f) .....

f1==f2 .....

∗p3 .....