Redefinition of the Nassi-Scheiderman STRUCT macro.00 ¡00

Übungen zu
Grundlagen der Informatik für Studierende des Maschinenbaus
im Sommersemester 2001
(Dipl.-Inf. Jörg Barner)

Aufgabe 13: Ablaufsteuerung
Welche Konstrukte zur Ablaufsteuerung kennen Sie? Geben Sie für jedes Konstrukt die Semantik und ein Beispiel in `C` an.

Lösung:
Sequenz: Nacheinanderausführung der Anweisungen. Beispiel:

```
a = 666;
b = 17;
```

unbedingter Sprung: In jedem Fall Verzweigung zu einem gegebenen Sprungziel. In `C`:

```
  goto marke;
  /* ... */
marke:
  /* ... */
```

Die Verwendung von `goto`'s ist nicht zu empfehlen, da sie den Überblick über die Ablaufstruktur stark erschweren! ("Spaghetticode")
Unterprogrammaufruf: Verzweigung zu einem Unterprogramm, nach dessen Beendigung direkt nach dem Aufruf fortgefahren wird. In `C` sind dies Funktionsaufrufe:

```
int f( int x ) { return( x + 1); }
/* ... */
a = f( b );
```

Alternative: Ausführen von Programmteilen in Abhängigkeit einer Bedingung. In `C`: die `if`-Anweisung (mit oder ohne `else`):

```
if (x != 0)
    y = a / x;
else
    y = 0;
```

Auswahl: Auswahl einer Alternative von vielen (mehr als zweien) durch Vergleich eines Ausdrucks mit verschiedenen Konstanten. Kann als Erweiterung der ein/zweiseitigen Altenative gesehen werden. In `C` die `switch`-Anweisung:

```
switch( n ) {
   case  0 : return( 1 );
   case  1 : return( 4 );
   case  2 : return( 16 );
   default : return( n - 3 );
}
```

Zählschleifen: Ein Programmabschnitt wird mehrfach durchlaufen, die Anzahl der Wiederholungen steht vor Beginn der Schleife fest. Meist wird für jeden Durchlauf des Schleifenkörpers eine bestimmte Variable mit einem neuen Wert belegt ("Laufvariable"). Beispiel:

```
for( i = 0; i < 10; ++i )
   printf( "%d\n", i );
```

Bedingungsschleifen: Ein Programmteil wird sooft durchlaufen bis eine Bedingung wahr oder falsch wird. Die Anzahl der Durchläufe ist somit i.a. nicht vorher bekannt, und es besteht die Gefahr von Endlosschleifen! Es gibt zwei Varianten dieses Schleifentyps: Bei der abweisenden Schleife wird die Bedingung geprüft, bevor der Körper das erste mal durchlaufen wird, es ist also möglich, daß dieser überhaupt nicht ausgeführt wird. In `C` ist dies die `while`-Schleife:

```
while( !feof(stdin) )
   tu_was( getc() );
```

Die nicht abweisende Schleife prüft die Bedingung erst nach dem Körper, er wird also immer mindestens einmal durchlaufen. In `C` ist dies das `do-while`-Konstrukt:

```
do {
    x = f(x);
} while( x > 0 );
```

In `PASCAL` wird `REPEAT-UNTIL` mit negierter Bedeutung der Bedingung benutzt. Das Äquivalent zu obigem Beispiel wäre also:

```
REPEAT
   x := f(x)
UNTIL x <= 0;
```

Aufgabe 14: Rekursion

a) Geben Sie ein Nassi-Schneiderman Diagramm für die rekursive Berechnung des größten gemeinsamen Teilers `ggt(a,b)` zweier nichtnegativer ganzer Zahlen `a` und `b` an. Der `ggt` ist definiert durch:

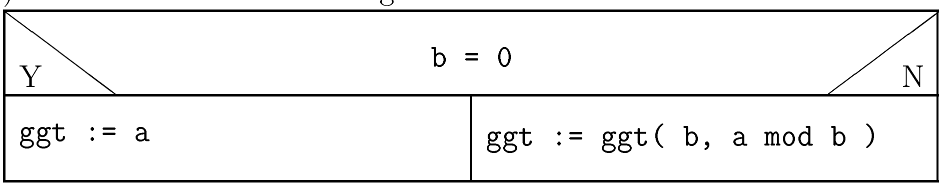
$$\mathtt{ggt(a,b)} = \begin{cases} \mathtt{a} & \text{für } \mathtt{b=0} \\ \mathtt{ggt(b,\ a\ mod\ b)} & \text{sonst} \end{cases}$$

b) Verfolgen Sie die Berechnung von 2^5 durch die unten angegbene rekursive Funktion `exp`, indem Sie für jeden Rekursionsschritt an den Programmstellen `A`, `B` und `C` die Werte von `n`, `b` und `ret` (jeweils sofern definiert / benutzt) notieren.

```
double exp( double a, unsigned n ) /* berechnet a^n */
{
   double b, ret;
   /* A */
   if (n == 0) { /* a^0 = 1 */
      ret = 1;
   } else if ((n % 2) == 1) {
      /* n ungerade: a^n = a * (a^(n-1)) */
      ret = a * exp( a, n-1 );
   } else {
      /* n gerade: a^n = a^(n/2) * a^(n/2) */
      /*     spart Rekursionsschritte      */
      b = exp( a, n/2 );
      /* B */
      ret = b * b;
   }
   /* C */
   return( ret );
}
```

Lösung:

a) Nasssi-Shneidermann Diagramm:



b) Berechnung von 2^5:

Rek.-tiefe	Schritt	`n`	`b`	`ret`
0	`A`	5	–	–
1	`A`	4	–	–
2	`A`	2	–	–
3	`A`	1	–	–
4	`A`	0	–	–
4	`C`	0	–	1
3	`C`	1	–	2
2	`B`	2	2	–
2	`C`	2	2	4
1	`B`	2	4	–
1	`C`	4	4	16
0	`C`	4	–	32