Universität Mannheim
Lehrstuhl für Praktische Informatik IV
Professor Dr. W. Effelsberg
Stephan Fischer, Raum 415

Multimedia-Technik
Wintersemester 1996/97
Übungsaufgaben

Multimediasysteme: Übungsblatt 3

Aufgabe 1 Lempel-Ziv

Betrachten Sie eine Zeichenkette der Länge 30. Entwickeln Sie eine Formel, die die maximale Größe des Dictionary in Abhängigkeit vom Umfang des verwendeten Alphabets angibt. Verwenden Sie diese Formel, um einen Algorithmus zu entwickeln, der zur Laufzeit entscheidet, ob die Verwendung von Lempel-Ziv effizient ist.

Hinweis: Betrachten Sie zuerst Alphabete der Länge 1 und 2

Lösung:

Sei ein Alphabet A gegeben, a sei die Länge des Dict.

Fall $\|A\| = 1$: Bedingung: $\sum_{i=1}^{a} i \leq 30$ ergibt a=7.

Fall $\|A\| = 2$: 2 1-er Kombinationen, 4 2-er Kombinationen und 6 3-er Kombinationen ergeben eine Stringlänge kleiner als 30. Es ergibt sich also a=11.

Allgemeine Formel:

Zu berechnen ist die Anzahl möglicher Permutationen, die dann den Einträgen im Dictionnary entsprechen. Diese ist $\sum_{i=1}^{n} \|A\|^i \cdot i$

.

Kodieralgorithmus: Prüfe die Länge des Dict. mit obiger Formel, falls Länge größer als 8 Bit, verwende z.B. Huffman.

Aufgabe 2 Dynamische arithmetische Kodierung

Die folgende Tabelle gibt die Wahrscheinlichkeiten für ein Alphabet {a,b,c} an. Sie wissen, daß diese Tabelle zur Dekompression der Zahlen *0.06 0.92* benutzt werden soll. Diese Zahlen wurden dynamisch arithmetisch kodiert. Jeder Zeichenblock ist 3 Zeichen lang. Wie lautet der Originalstring?

Buchstabe	Wahrscheinlichkeit
a	0.3
b	0.3
c	0.4

Lösung:

Dekodiere 0.06:

0.06 ist in Intervall 0.054-0.09 ==> C
0.054 ist in Intervall 0-0.09 ==> A
0.09 ist in Intervall 0-0.3 ==> A

Also Zeichenkette AAC

Errechne neue Tabelle:

Buchstabe	Wahrscheinlichkeit
a	0.48
b	0.15
c	0.37

0.92 ist in Intervall 0.8631-0.9288 ==> A
0.9137 ist in Intervall 0.8631-1 ==> C
0.8631 ist in Intervall 0.63-1 ==> C

Kodierter String: AAC CCA

Aufgabe 3 Arithmetische Kodierung

Programmieren Sie einen Algorithmus, der einen String arithmetisch kodiert. Die Tabelle der Wahrscheinlichkeiten sei fest vereinbart.

Lösung:

```
float prob[5], probCum[5], oldVal;

...

prob[0]=0.3; prob[1]=0.1; prob[2]=0.15; prob[3]=0.15; prob[4]=0.3;
probCum[0]=0.3; probCum[1]=0.4; probCum[2]=0.55; probCum[3]=0.7; probCum[4]=1.0;


float arithCod(int interval, float *left, float *right)
{
/* use oldval to store the interval width */
   float result=1;
   oldVal *=prob[interval];
   *left+=probCum[interval]*oldVal;
   *right=*left+oldVal;
   result=*right-*left;
   return 0;


}



main(int argc, char** argv)
{
   int i=0, step=0;
    float result=1, left=0, right=1;
    oldVal=1;
    while(argv[1][i++]!='\ 0')
{
       switch(argv[1][i])
       case 'a':
            result =arithCode(0,&left,&right);
            break;
       switch(argv[1][i])
       case 'b':
```

```
            result =arithCode(1,&left,&right);
            break;
        switch(argv[1][i])
        case 'c':
            result =arithCode(2,&left,&right);
            break;
        switch(argv[1][i])
        case 'd':
            result =arithCode(3,&left,&right);
            break;
        switch(argv[1][i])
        case 'e':
            result =arithCode(4,&left,&right);
            break;

}

printf(" %f ",result);
}
```