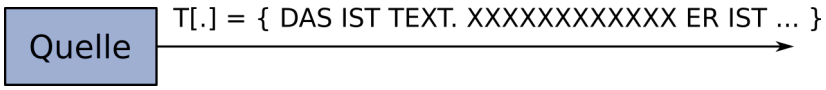
Dr. Jürg M. Stettbacher
Neugutstrasse 54
CH-8600 Dübendorf

Telefon: +41 43 299 57 23
Email: dsp@stettbacher.ch

Übung

Run Length Encoding (RLE)



Wir betrachten eine Quelle, die Zufallsvariablen $T[n]$ erzeugt. Jedes $T[n]$ kann ein Symbol aus der Menge aller Grossbuchstaben, dem Leerzeichen und dem Punkt wiedergeben, die je mit 5 Bit codiert sind. Eine Symbolfolge $T[.]$ von derartigen Zufallsvariablen bildet deutschen Text, wie oben dargestellt. Im Text können Sequenzen, resp. Runs von Füllzeichen vorkommen (in der Abbildung sind die Füllzeichen X), die häufig zwischen 40 und 130 Zeichen lang sind.

1. Ist die Quelle gedächtnislos oder nicht?
2. Schlagen sie einen RLE für diese Quelle vor.
3. Wenden Sie Ihren Vorschlag auf den oben angegebenen Ausgang der Quelle an (vom ersten Zeichen D bis zum letzten T). Wie lautet die komprimierte Symbolfolge?
4. Welche Kompressionsrate[1] R ergibt sich bei der gegebenen Symbolfolge?
5. Komprimieren Sie diese Folge:

   ABCDEFGHIJKLMNOPQRSTUVWXYZ._

   Beachten Sie, dass das letzte Symbol das Leerzeichen meint.
6. Notieren Sie den Algorithmus, der Ihre RLE codierte Symbolfolge decodiert.

[1] Die Kompressionsrate R ist definiert als die Anzahl Bits am Ausgang des Encoders dividiert durch die Anzahl Bits am Eingang.

Antworten

1. Die Quelle hat ein Gedächtnis. Aufeinanderfolgende Symbole sind in der deutschen Sprache nicht unabhängig von einander. Beispielsweise folgt auf einen Buchstaben Q fast immer ein U. Oder auf ein C folgt fast immer ein H oder ein K. Der Vorgängerbuchstabe hat also einen Einfluss auf den Nachfolger.

2. Vorschlag: Eine Sequenz wird ersetzt durch einen Token der Gestalt (M, L, Z). Beachten Sie, dass Klammern und Kommas hier nur informativen Charakter haben um den Token darzustellen.

   (a) M ist ein Marker für den Start des Tokens. Wir wählen mit Vorteil ein selten auftretendes Zeichen. Tritt nämlich das Zeichen im Text auf, so muss es - um eine Verwechslung zu vermeiden - durch einen Token ersetzt werden. Wir wählen beispielsweise $M = \mathrm{Y}$. Die Buchstaben X oder Q wären ebenfalls gute Kandidaten.

   (b) L ist die Länge der Sequenz, resp. des Runs. L ist eine binäre Zahl fester Breite, damit der Decoder weiss, wieviele Bits er nach dem Marker dafür lesen muss. Aufgrund der typischen Länge der Runs ist eine Breite von 7 Bit zweckmässig. Damit können zwar nur Runs bis zur Länge 127 komprimiert werden. Aber ein zusätzliches Bit für L lohnt sich nicht aufgrund der wenigen Runs, die länger als 127 Zeichen sind.

   (c) Z ist das Symbol, das im Run wiederholt wird.

3. Die komprimierte Sequenz sieht so aus:

   DAS IST TEXT. Y(12)X ER IST

   Beachten Sie: Die Klammer ist nicht Teil der Symbolfolge, sondern markiert den Zähler $L = 12$, der im Gegensatz zu den Quellensymbolen eine eigene Bit-Breite hat.

4. Ein Token umfasst $5 + 7 + 5 = 17$ Bit. Der Quellenausgang zählt 33 Symbole an je 5 Bit. Die komprimierte Symbolfolge umfasst noch 21 Symbole an je 5 Bit plus einen Token an 17 Bit. Damit folgt die Kompressionsrate:

   $$R = \frac{21 \cdot 5 + 17}{33 \cdot 5} = 0.739$$

   Es resultiert also eine Kompression.

5. Komprimierte Folge:

   ABCDEFGHIJKLMNOPQRSTUVWXY(1)YZ._

   Das Symbol Y, das gleichzeitig als Marker M dient, muss in einen Token mit $L = 1$ (in Klammer dargestellt) und $Z = \mathrm{Y}$ umgewandelt werden.

6. Algorithmus für den Decoder:

   (a) Versuche, das nächste Zeichen (5 Bit) zu lesen.
   Falls kein Zeichen gelesen werden kann: $\longrightarrow$ Ende.

   (b) Falls das gelesene Zeichen kein Marker $M = \mathrm{Y}$ ist,
   Gib das gelesene Zeichen aus.
   Dann: $\longrightarrow$ (a)

   (c) Lies den Zähler L (7 Bit).
   Falls Lesen nicht möglich: $\longrightarrow$ Fehler.

   (d) Lies das Zeichen Z (5 Bit).
   Falls Lesen nicht möglich: $\longrightarrow$ Fehler.

   (e) Gib das Zeichen Z genau L mal aus.
   Dann: $\longrightarrow$ (a)