

# Lempel-Ziv-Welch Verfahren

---

## 1. Thema: Lempel-Ziv-Welch Verfahren

## 2. Vorwort:

Herzlich willkommen zu diesem Tutorial,

ich freue mich, dass du dich für eines meiner Dokumente entschieden hast. In diesem kleinen Tutorial möchte ich das Lempel-Ziv-Welch Verfahren etwas beleuchten und die Funktionsweise dieses Kompressionsverfahren näher erläutern. Da im deutschsprachigen Raum kaum brauchbare Dokumentationen zu diesem Algorithmus erhältlich sind, hoffe ich, dass meine Erklärungen zumindest das grundlegende Verständnis von LZW Kompression fördern.

In diesem Sinne viel Spaß beim Lesen!

## 3. Geschichte:

Der Lempel-Ziv-Welch Algorithmus basiert auf einem Kompressionsverfahren, das Abraham Lempel und Jacob Ziv im Jahre 1978 unter dem Namen LZ78 veröffentlicht haben und das 1984 von Terry Welch verbessert wurde. Es ermöglicht verlustfreie Datenkompression und wird deshalb bevorzugt in Grafikformaten verwendet. Das 1987 von CompuServ entwickelte GIF Format basiert beispielsweise auf dem LZW Algorithmus, aber auch JPEG und TIFF Dateien können mit diesem Verfahren komprimiert werden.

Das Verfahren war lange Zeit in den Schlagzeilen, weil die Sperry Corporation (später Unisys Corporation) in den 1990er Jahren Patente an eben diesem angemeldet hatte und fortan Lizenzgebühren für die Verwendung des GIF Formates berechnete. Die rechtliche Gültigkeit dieser Patente auf einen Algorithmus war jedoch Jahre lang umstritten, bis sie im Jahr 2004 letztendlich unverlängerbar ausliefen.

## 4. Funktionsweise:

Bevor wir uns die Funktionsweise des LZW Verfahrens genauer ansehen, sollten wir klären, welche Grundidee überhaupt hinter Kompression steckt. Grundsätzlich gehen wir dabei von Dateien aus, die unabhängig von ihrem Typ (Texte, Bilder, Musik, Filme) aus Daten bestehen. Der zu packende Datenstrom besteht dabei wiederum aus einzelnen Zeichen, in denen gewisse Zeichenfolgen unregelmäßig immer wieder auftauchen. Identische Vorkommnisse können nun zusammengefasst und mit geringerem Speicherbedarf wieder abgespeichert werden.

Eine Möglichkeit der Datenkompression ist nun das Lempel-Ziv-Welch Verfahren, dass man auch als wörterbuchbasierten Algorithmus bezeichnet. Der zu komprimierende Datenstrom wird bei diesem Verfahren in einzelne Zeichenketten (Strings) zerteilt, die nach und nach in einem Wörterbuch (auch Dictionary oder Stringtable genannt) abgelegt werden.

Während dieses Ablegevorgangs wird immer wieder geprüft, ob die aktuelle Zeichenkette sich bereits im Wörterbuch befindet. Ist dies der Fall, so wird an deren Stelle im Datenstrom nur noch ein Verweis auf den entsprechenden Eintrag im Stringtable hinterlassen. Ist der

String hingegen noch nicht im Wörterbuch vorhanden, so wird er am Ende als neuer Eintrag angefügt.

Betrachten wir nun einen kurzen Datenstrom, um das Verfahren zu verdeutlichen:  
RRGBRRBGBBR

Diesen zerlegen wir in verschiedene Zeichenketten:  
RR GB RR B GB BR

Stellen wir uns nun vor, dass ein LZW Algorithmus die Zeichenketten nach und nach überprüft. Mehrfach vorkommende Zeichenfolgen würden im Dictionary (hier in eckigen Klammern dargestellt) abgelegt und durch einen Verweis auf den entsprechenden Eintrag ersetzt.

121B2BR [1=RR, 2=GB]

Dies wäre bereits eine einfache Form von Datenkompression. Je länger die einzelnen Zeichenketten (auch Sequenzen genannt) nun werden, umso effektiver die Kompression. Je kürzer die Zeichenketten, umso größer hingegen der Speicherbedarf. Betrachten wird deshalb ein weiteres Beispiel:

GGRBRRRBGRBBRBGGRBRBRBRBBBGGBRBR

#### 1. Variante:

GG RBRB R RBG RB B RBG G RBRB RBRB BB GG B RB R

Komprimiert:

12R34B3G22BB1B4R [1=GG, 2=RBRB, 3=RBG, 4=RB]

#### 2. Variante:

GGRBRBR RB G RB B RB GGRBRB RB RB BBGGB RB R

Komprimiert:

12G2B2122BBGGB2R [1=GGRBRBR, 2=RB]

Wie man unschwer erkennen kann, wird der Speicherbedarf der zweiten Variante weitaus geringer sein, da das Dictionary weniger Einträge enthält. Aufgabe des LZW Verfahrens ist es somit, den zu komprimierenden Datenstrom in möglichst wenige, aber dafür lange und häufig vorkommende Sequenzen zu zerlegen.

#### 5. Nachwort:

Ich hoffe, dass meine Erklärungen verständlich waren und dir dieses Tutorial weitergeholfen hat. Für Kritik, Lob und Verbesserungsvorschläge per E-Mail bin ich immer dankbar.

Mit freundlichen Grüßen  
*Sascha Seidel aka NetPanther*

**E-Mail:** webmaster@SaschaSeidel.de

**Website:** <http://www.SaschaSeidel.de>