Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering

Kurs Information und Codierung

Datenkompression nach Lempel Ziv (LZ)
Übersicht

Studiengang IT
21.08.2017

https://olat.zhaw.ch/olat/...

Autor: Kurt Hauser husr@zhaw.ch

Dozenten: Dr. Jürg Stettbacher, Kurt Hauser

Lernziele

- **Die Studierenden kennen die Bedeutung und den Anwendungsbereich der *Lempel Ziv Codierung***
- **Sie erhalten eine Übersicht der Varianten der Lempel Ziv Codierung**
- **Sie sind in der Lage, anhand spezieller Bedürfnisse die verschiedenen Verfahren grundsätzlich einander gegenüberzustellen.**

Lempel Ziv: Übersicht

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering

Diese Übersicht enthält Angaben zu:

- LZ77
- LZSS, LZX
- LZ78
- LZFG, LZRW1
- LZW
- LZMW, LZAP, LZP
- Literaturangaben

Lempel Ziv Übersicht: Allgemeines

- Die Lempel Ziv Codierung gehört zum Bereich der Quellencodierung
- Mittels der Lempel Ziv Codierung werden Token erzeugt, aus denen der ursprüngliche Text respektive die ursprüngliche Zeichenkette exakt und ohne Verluste wieder hergestellt werden kann
- Die Darstellung durch die Token führt in den meisten Fällen zu einer Datenkompression

Lempel Ziv Übersicht: Allgemeines

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering

Der Aufwand für die Encodierung d.h. Berechnung der Token hängt von der Wörterbuchgrösse, der Grösse des Vorschaubuffers und von der Art der Codierung ab.

Die Leistungsfähigkeit der Lempel Ziv Codes ist schon bei kleinen mittleren Wortgrössen (z.B. 5 Bytes) und Wörterbuchgrösse von rund 500’000 Wörtern gegeben, vgl. Unterabschnitt über ‘Datenkompression mit statischem Wörterbuch’.

Lempel Ziv Übersicht: Allgemeines

Zürcher Hochschule
für Angewandte Wissenschaften
zh aw School of Engineering

Bei der Lempel Ziv Codierung wird grundsätzlich eine Kette von Zeichen in Relation mit einer bekannten Zeichenkette gesetzt.

Die verschiedenen Verfahren, die aus dem Hauptverfahren «Lempel Ziv Codierung» abgeleitet sind, werden im Folgenden im Sinne einer Übersicht vorgestellt.

LZ77

- **LZ77 ist ein ‘Sliding Window’ Verfahren**
- **Symbole, die im Vorschaubuffer auftreten, werden im Suchbuffer gesucht.**
- **Der codierte Text nimmt eine Form wie folgt an: (17,3,»<Zeichen x>»), (21,2,»<Zeichen y>»), etc.**
- **Das Verfahren ist im Script detailliert beschrieben; es existiert eine Übung zu LZ77.**

LZSS

LZSS ist eine durch James A. Storer und T.G. Szymanski geänderte Version des LZ77-Verfahrens.

- **1. Änderung gegenüber LZ77: Der Suchbuffer («the dictionary») wird als binärer Suchbaum gebildet**
- **2. Änderung gegenüber LZ77: Die Token enthalten nur zwei Felder anstatt drei**

LZX

- **Bei LZX ist der Umstand genutzt, dass sich bestimmte Werte des ‘Offset’ wiederholen.**
- **Beispiel: Eine Zeichenkette werde zum Token (23, <Länge>) komprimiert. Es besteht eine bestimmte Wahrscheinlichkeit p, dass der Offset 23 wieder vorkommt für eine andere (spätere) Zeichenkette.**
- **Drei Spezialcodes 0, 1 und 2 werden genutzt, um die drei häufigsten Offsetwerte zu bezeichnen.**

LZ78

- **Basis für das LZ78-Verfahren ist ein Wörterbuch, welches zu Beginn leer ist und dann laufend angereichert wird.**
- **Kein ‘Sliding Window’ Verfahren**
- **Der Token bei LZ78 nimmt folgende Form an: (Pointer auf String im WB, «neues» Symbol)**
- **Das Verfahren ist im Script detailliert beschrieben; es existiert eine Übung zu LZ78**

LZFG

LZFG ist eine durch Edward Fiala und Daniel Greene beschriebene LZ-Variante

- **Der Encoder erzeugt ein komprimiertes File, welches aus Token (2 Typen) sowie Einzelzeichen (ASCII Zeichen) besteht**
- **Token können ihrerseits sein:**
  Literal Token (zeigt auf neue Zeichenkette)
  Copy Token (zeigt auf einen bereits bekannten String)

LZRW1

LZRW1 ist eine durch Ross Williams beschriebene LZ Variante.

- **Einfache, ‘schnelle’ Variante von LZ77**
- **Grundidee: Eine Übereinstimmung in nur einem Schritt finden, mittels Silbentabelle (hash table)**
- **Schnell, jedoch nicht sehr effizient, da die gefundene Übereinstimmung oftmals nicht die längstmögliche ist.**

LZRW4

Ebenfalls auf der Idee von Ross Williams basierte LZ Variante.

- **LZRW4 nutzt einen 1 Mbyte grossen Buffer**
- **Bei jedem Schritt des Codierprozesses ist ein Kontext 2. Ordnung benutzt: Die zwei letzten Symbole im Suchbuffer werden benutzt, um das neue Symbol vorauszusagen.**
- **Für Details Siehe [1]**

LZW

LZW ist eine Variante des LZ78 Verfahrens

- Das Symbol-Feld im Token ist eliminiert
- Als gemeinsame Startbasis für Sender und EMpfänger dient die ASCII Tabelle
- Ein Token besteht nur noch aus einem Zahlenwert, Beispiele: 65, 78, 256, 65, 83
- Das LZW-Verfahren ist im Script detailliert beschrieben; es existiert eine Übung zu LZW.

LZMW

LZMW ist eine von V. Miller und M. Wegmann beschriebene LZ-Variante

LZMW basiert auf zwei Prinzipien:

- **Sobald das Wörterbuch (WB) voll ist, wird der am wenigsten genutzte Wörterbucheintrag aus dem Wörterbuch gelöscht**
- **Jede Silbe, die dem WB beigefügt wird, ist die Verkettung zweier Strings.**

LZAP

LZAP ist eine Erweiterung von LZMW

‘AP’ steht für «all prefixes»

- Anstatt jeweils zwei Strings zu verketten, werden zusätzliche Verkettungen gemacht:
- Beispiel: X = a, X’ = qzx: Damit werden aq, aqz sowie aqzx allesamt zum Wörterbuch beigefügt.

LZP ist eine Variante von LZ77

- Beschrieben von Charles Bloom. Das «P» steht für ‘prediction’.
- Idee: Falls ein bestimmter String ‘ung’ im Zeichenstrom vorkam und dieser String dann mit ‘en’ weiterging, dann ist die Wahrscheinlichkeit hoch, dass ein wiederum vorkommendes ‘ung’ abermals mit ‘en’ weitergeht.

Quellenverzeichnis

[1] ‘Data Compression’, David Salomon; Springer Auflage 2004; ISBN 0-387-40697-2

[2] ‘Information und Kommunikation’, Markus Hufschmid; Teubner Verlag; Auflage 2006; ISBN 978-3-8351-0122-7

Fragen