

Grundlagen der Informatik

Prof. Dr. J. Schmidt

Fakultät für Informatik

GDI – WS 2020/21

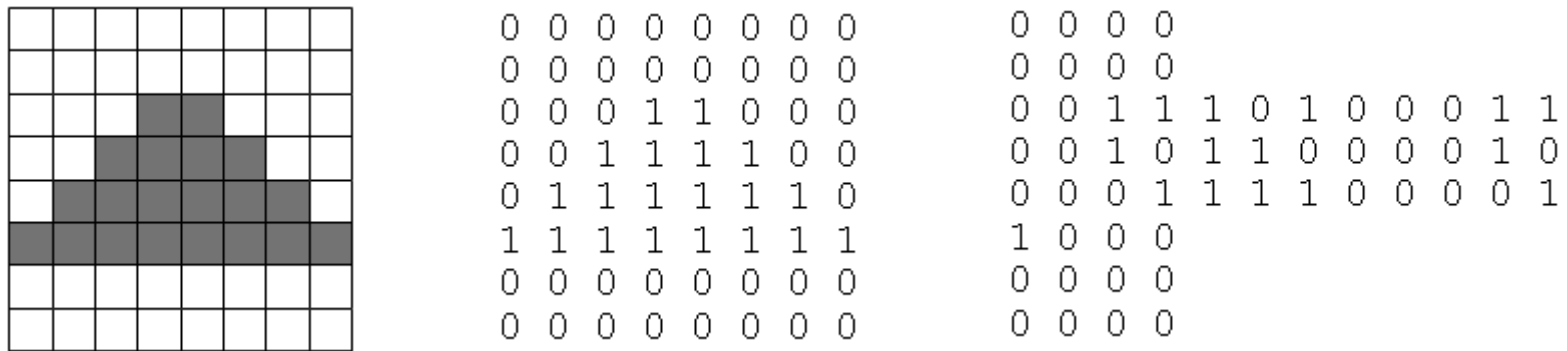
Information und Quellencodierung
Lauf längencodierung / LZW-Kompression

- „RLE“ / „RLC“
 - Engl. „run-length-encoding“ / „run-length-coding“
- Eindimensionale Laufängen-Codierung
 - Ergänzung eines Zählers
 - zusätzlich zu den codierten Daten
- Speicherung
 - Laufänge = Anzahl der Wiederholungen eines Zeichens
 - Speicherung von Zahlenpaaren etwa der Art (f,n)
 - f gibt den Datenwert an und
 - n gibt die Laufänge an
(gerechnet ab Anfang des Datenstroms bzw. ab Ende der vorherigen Sequenz)



Einfaches Beispiel

● Lauflängen-Codierung eines Binärbildes



- Übertragung Zahlenpaare (Datenwert, Lauflänge)

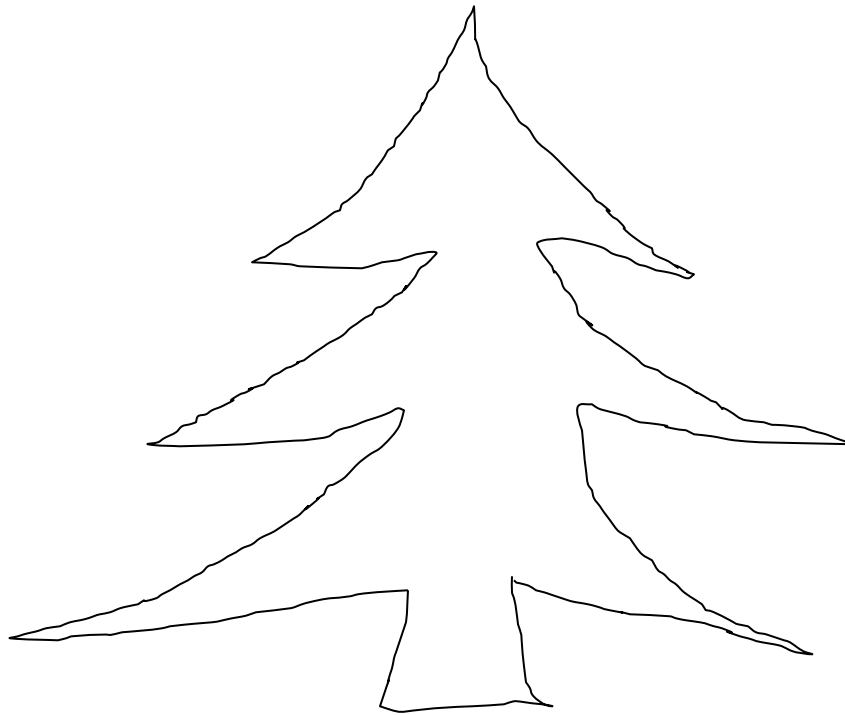
- Lauflängen: 001=1 010=2 011=3 100=4
101=5 110=6 111=7 000=8

- Kompression von 64 auf 56 Bit wird erreicht



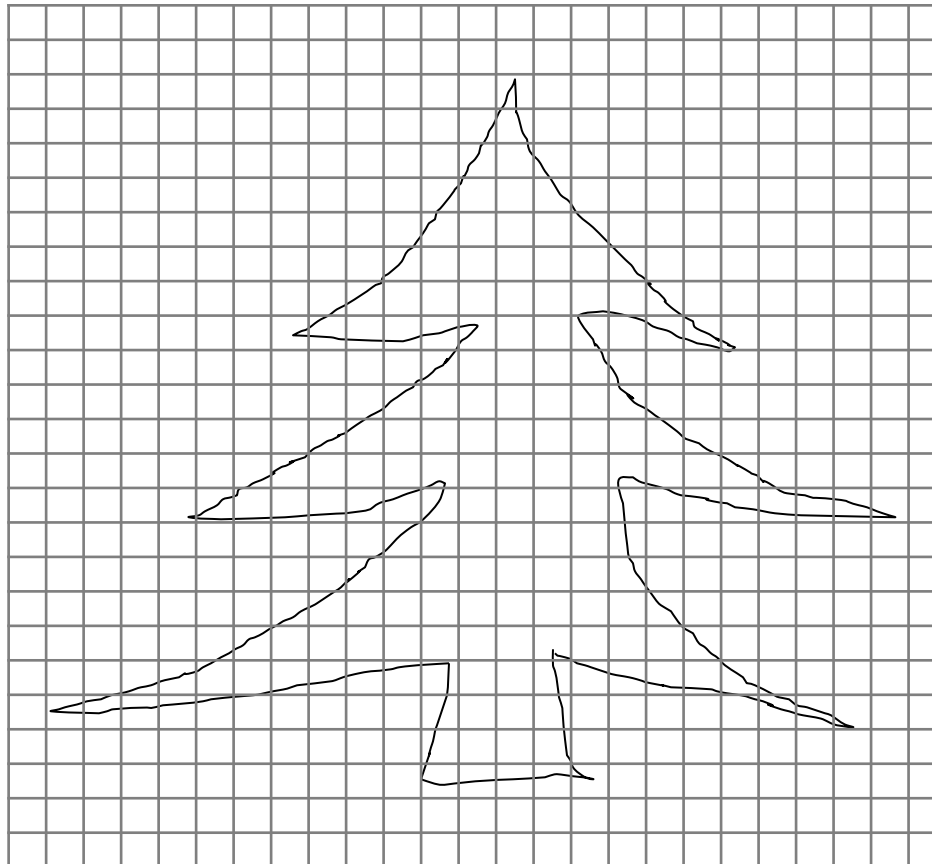
Anwendungsbeispiel: Fax schicken

Original



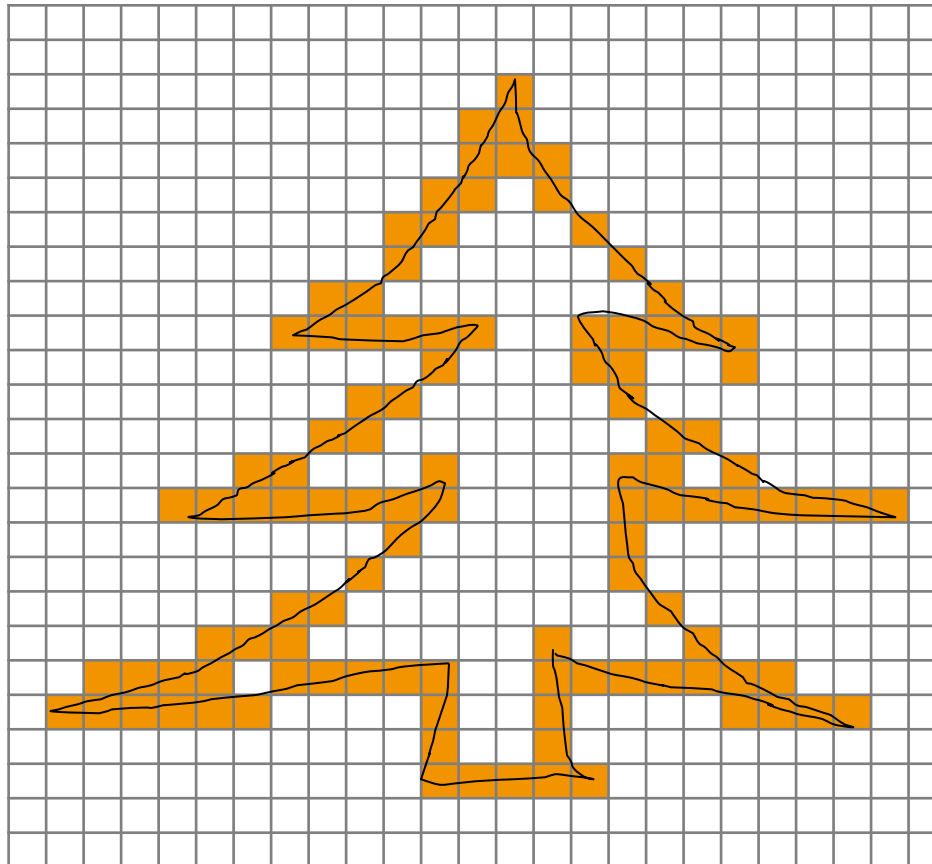
Anwendungsbeispiel: Fax schicken

Gitternetz



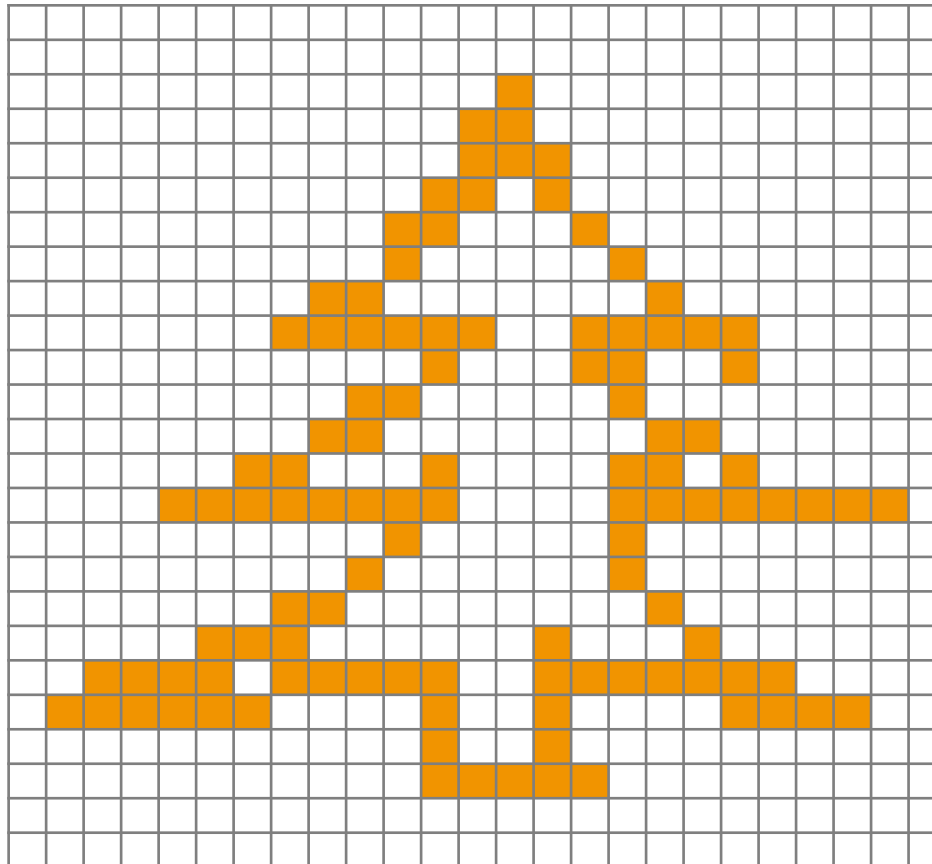
Anwendungsbeispiel: Fax schicken

Rasterbild



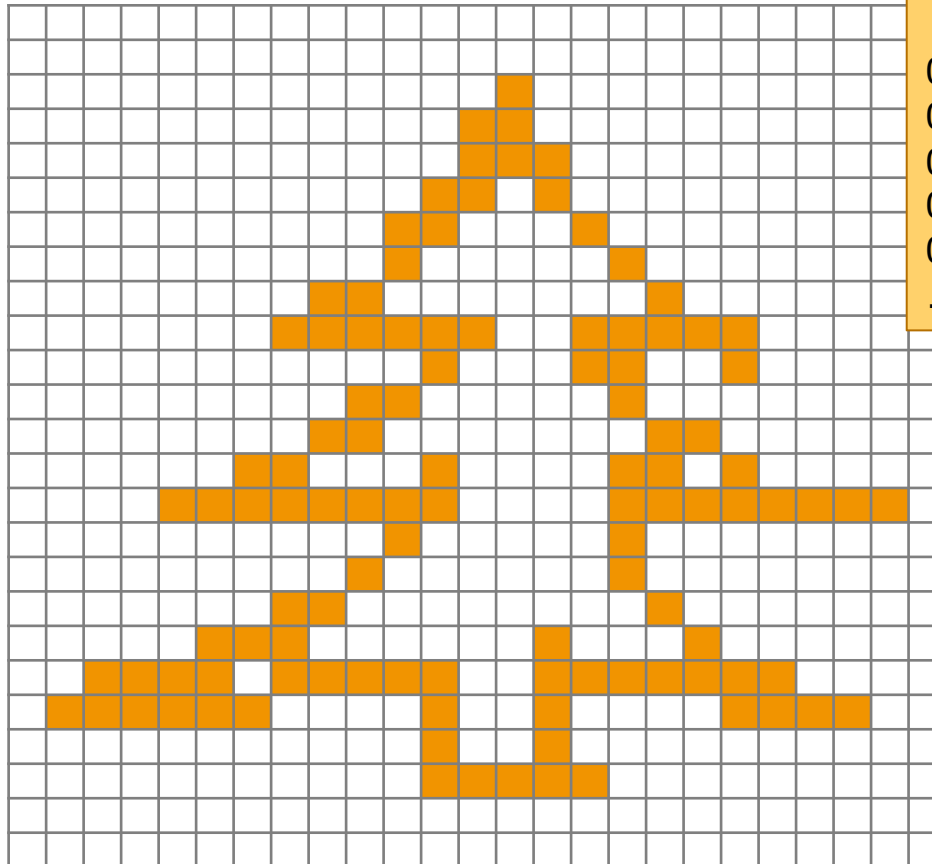
Anwendungsbeispiel: Fax schicken

Rasterpunkt
=
Speicherstelle



Anwendungsbeispiel: Fax schicken

625 Bit



0 = weiß
1 = schwarz

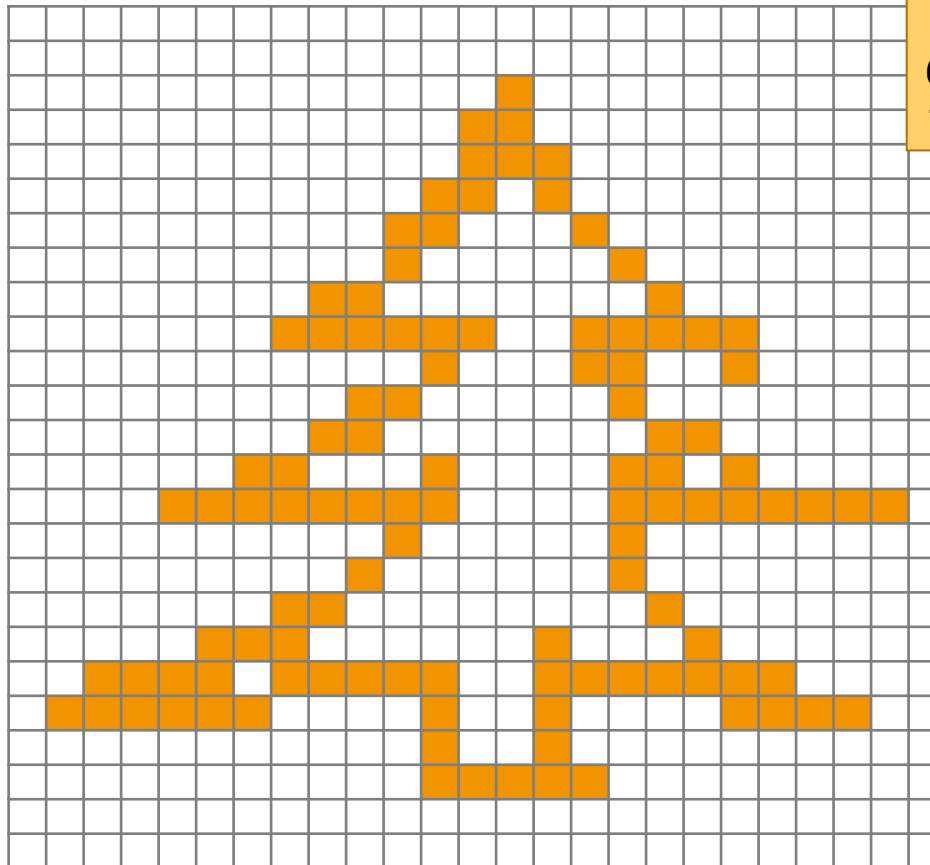
```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000010000000000000000
00000000000001100000000000000000
00000000000001110000000000000000
```

□ □



Anwendungsbeispiel: Fax schicken

Laufängen



0 = weiß
1 = schwarz

63 1 23 2 23 3 21 2 1 1 20 2 3
1 19 1 5 1 16 2 7 1 ...

Anwendungsbeispiel: Fax schicken

- Laufängen

63 1 23 2 23 3 21 2 1 1 20 2 3 1 19 1 5 1 16 2 7 1 ...

- Weitere Verbesserung

- Berücksichtigung der Häufigkeiten der Laufängen
- Code-Baum mit variabler Code-Länge

- Faxgeräte

- Eingespeicherte „Standard Code-Bäume“



Beobachtungen

- **Effiziente** Komprimierung nur möglich,
 - falls in den Daten **zahlreiche homogene Bereiche** auftreten,
 - die durch ein einziges Codewort charakterisiert werden können
- Anwendung v.a. in
 - computergenerierten Bildern und Grafiken
 - sowie Binärbildern mit zwei Helligkeitsstufen
 - Kombination mit Huffman (z.B. in JPEG)
- Vergrößerung der Datei möglich
 - Bei wenigen längeren Sequenzen von identischen Daten



- Erfinder des Algorithmus

- Lempel, Ziv und Welch (1978/83)

- Zugrunde liegende Idee

- Erweiterung RLC:
Nicht nur Codierung von Einzelzeichen, sondern auch von Zeichengruppen unterschiedlicher Länge
 - Berücksichtigung von Häufigkeiten der Einzelzeichen sowie Redundanzen, die auf der Korrelation aufeinander folgender Zeichen beruhen
 - Erkennen von redundanten Zeichenfolgen, die durch kürzeren Code ersetzt werden



Ergebnis

- LZW-Algorithmus **minimiert** die **Redundanzen**, die dadurch entstehen, dass identische Zeichenfolgen sich in den Eingabedaten mehrmals wiederholen
- Die **Kompressionswirkung** ist **umso besser**, **je häufiger** solche **Wiederholungen** auftreten und **je länger** die sich **wiederholenden Zeichenfolgen** sind
- Ergebnis ist eine weitgehend unkorrelierte Zeichenfolge, die verlustfrei nicht mehr weiter komprimierbar ist



Prinzip

- **Dynamische Generierung** einer Code-Tabelle
- Jeder Eintrag besteht aus
 - einer Zeichenfolge mit Zeichen aus dem Quell-Alphabet
 - und dem zugehörigen komprimierten Code
- Code-Tabelle
 - wird zu Beginn mit allen Einzelzeichen des Quell-Alphabets vorbesetzt
 - wird während der Kompression nach und nach erweitert und an die Eingabe angepasst



Charakteristische Eigenschaften

- Benötigt keine Informationen über die Statistik des Eingabetextes
 - (z.B. Auftrittswahrscheinlichkeiten der Einzelzeichen)
- Code-Tabelle muss nicht zusammen mit den codierten Daten gespeichert bzw. übertragen werden
 - Wird bei Decodierung aus den codierten Daten in identischer Weise wieder erzeugt



Anwendung

- Bild- und Dokumentformate
 - Zum Sparen von Speicherplatz
- Beispiele
 - GIF, TIFF, PDF, Postscript



LZW-Kompression eines Strings Z

- Initialisiere die Code-Tabelle mit den Einzelzeichen
- Weise dem Präfix P den Leerstring zu
- Wiederhole, solange Eingabezeichen vorhanden sind:
 - Lies nächstes Eingabezeichen c aus dem Eingabestring Z
 - Wenn Pc in der Code-Tabelle gefunden wird:
 - Setze $P=Pc$
 - Sonst:
 - Trage Pc in die nächste freie Position der Code-Tabelle ein
 - Gib den Code für P aus
 - Setze $P=c$
- Ende der Schleife
- Gib den Code für das letzte Präfix P aus



Beispiel: Codierung der Zeichenkette $Z = \text{ABABCBABAB}$

- Vorbereitung
 - Initialisiere die Code-Tabelle mit den Einzelzeichen

Präfix	Code
A	0
B	1
C	2



Beispiel: Codierung der Zeichenkette $Z = \text{ABABCBABAB}$

● Codierungsvorgang

Zeichen c	Präfix P	Ausgabe
	-	
A	A	
B	B	0
A	A	1
B	AB	
C	C	3
B	B	2
A	BA	
B	B	4
A	BA	
B	BAB	
		7

Code-Tabelle

Präfix	Code
A	0
B	1
C	2
AB	3
BA	4
ABC	5
CB	6
BAB	7

Lies nächstes Eingabezeichen c
Wenn (Pc in der Code-Tabelle)

Setze $P=Pc$

Sonst

Trage Pc in Code-Tabelle ein

Gib den Code für P aus

Setze $P=c$

Gib den Code für
das letzte Präfix P aus

➔ Codierte Nachricht: **013247**



LZW-Dekompression einer Nachricht

- Initialisiere die Code-Tabelle mit den Eingabezeichen
- Weise dem Präfix P den Leerstring zu
- Wiederhole, solange Eingabezeichen vorhanden sind:
 - Lies nächstes Eingabezeichen c
 - Wenn c in der Code-Tabelle enthalten ist:
 - Gib den zu c gehörenden String aus
 - Setze k = erstes Zeichen dieses Strings
 - Trage Pk in die Code-Tabelle ein, falls noch nicht enthalten
 - Setze P = zum Code c gehörender String
 - Sonst (Sonderfall):
 - Setze k = erstes Zeichen von P
 - Gib Pk aus
 - Trage Pk in die Code-Tabelle ein
 - Setze $P = Pk$
- Ende der Schleife



Beispiel: Dekomprimierung von 013247

● Decodierungsvorgang

Code c	k	Präfix P	Ausgabe
		-	
0	A	A	A
1	B	B	B
3	A	AB	AB
2	C	C	C
4	B	BA	BA
7	B	BAB	BAB

Code-Tabelle

Präfix	Code
A	0
B	1
C	2
AB	3
BA	4
ABC	5
CB	6
BAB	7

Lies nächstes Eingabezeichen c

Wenn (c in Code-Tabelle)

Gib den zu c gehörenden String aus

Setze k = erstes Zeichen dieses Strings

Trage Pk in Code-Tabelle ein, falls nicht enth.

Setze P = zum Code c gehörender String

Sonst

Setze k = erstes Zeichen von P

Gib Pk aus

Trage Pk in die Code-Tabelle ein

Setze $P = Pk$

➔ Decodierte Nachricht: **ABABCBABAB**



Codieren Sie die Zeichenkette ABBABABAC mit Hilfe des LZW-Algorithmus.

Lies nächstes Eingabezeichen c
Wenn (Pc in der Code-Tabelle)
 Setze $P=Pc$
Sonst
 Trage Pc in Code-Tabelle ein
 Gib den Code für P aus
 Setze $P=c$

Gib den Code für
das letzte Präfix P aus

