

CSE 490 GZ
Introduction to Data Compression
Winter 2004

Dictionary Coding
LZW

Dictionary Coding

- Does not use statistical knowledge of data.
- Encoder: As the input is processed develop a dictionary and transmit the index of strings found in the dictionary.
- Decoder: As the code is processed reconstruct the dictionary to invert the process of encoding.
- Examples: LZW, LZ77, Sequitur,
- Applications: Unix Compress, gzip, GIF

CSE 490gz - Lecture 7 - Winter 2004

2

LZW Encoding Algorithm

Repeat
find the longest match w in the dictionary
output the index of w
put wa in the dictionary where a was the
unmatched symbol

CSE 490gz - Lecture 7 - Winter 2004

3

LZW Encoding Example (1)

Dictionary a b a b a b a b a
0 a
1 b

CSE 490gz - Lecture 7 - Winter 2004

4

LZW Encoding Example (2)

Dictionary a b a b a b a b a
0 a
1 b
2 ab
0

CSE 490gz - Lecture 7 - Winter 2004

5

LZW Encoding Example (3)

Dictionary a b a b a b a b a
0 a
1 b
2 ab
3 ba
0 1

CSE 490gz - Lecture 7 - Winter 2004

6

LZW Encoding Example (4)

Dictionary ababa b a b a
0 a 0 1 2
1 b
2 ab
3 ba
4 aba

LZW Encoding Example (5)

Dictionary abababa b a
0 a 0 1 2 4
1 b
2 ab
3 ba
4 aba
5 abab

LZW Encoding Example (6)

Dictionary ababababa
0 a 0 1 2 4 3
1 b
2 ab
3 ba
4 aba
5 abab

LZW Decoding Algorithm

- Emulate the encoder in building the dictionary. Decoder is slightly behind the encoder.

```
initialize dictionary;  
decode first index to w;  
put w? in dictionary;  
repeat  
  decode the first symbol s of the index;  
  complete the previous dictionary entry with s;  
  finish decoding the remainder of the index;  
  put w? in the dictionary where w was just decoded;
```

LZW Decoding Example (1)

Dictionary 0 1 2 4 3 6
0 a a
1 b
2 a?

LZW Decoding Example (2a)

Dictionary 0 1 2 4 3 6
0 a a b
1 b
2 ab

LZW Decoding Example (2b)

Dictionary

0	a	0	1	2	4	3	6
1	b	a b					
2	ab						
3	b?						

CSE 490gz - Lecture 7 - Winter 2004

13

LZW Decoding Example (3a)

Dictionary

0	a	0	1	2	4	3	6
1	b	a b a					
2	ab						
3	ba						

CSE 490gz - Lecture 7 - Winter 2004

14

LZW Decoding Example (3b)

Dictionary

0	a	0	1	2	4	3	6
1	b	a b ab					
2	ab						
3	ba						
4	ab?						

CSE 490gz - Lecture 7 - Winter 2004

15

LZW Decoding Example (4a)

Dictionary

0	a	0	1	2	4	3	6
1	b	a b ab a					
2	ab						
3	ba						
4	aba						

CSE 490gz - Lecture 7 - Winter 2004

16

LZW Decoding Example (4b)

Dictionary

0	a	0	1	2	4	3	6
1	b	a b ab aba					
2	ab						
3	ba						
4	aba						
5	aba?						

CSE 490gz - Lecture 7 - Winter 2004

17

LZW Decoding Example (5a)

Dictionary

0	a	0	1	2	4	3	6
1	b	a b ab aba b					
2	ab						
3	ba						
4	aba						
5	abab						

CSE 490gz - Lecture 7 - Winter 2004

18

LZW Decoding Example (5b)

Dictionary

0	a	012436
1	b	a b ab aba ba
2	ab	
3	ba	
4	aba	
5	abab	
6	ba?	

CSE 490gz - Lecture 7 - Winter 2004

19

LZW Decoding Example (6a)

Dictionary

0	a	012436
1	b	a b ab aba ba b
2	ab	
3	ba	
4	aba	
5	abab	
6	bab	

CSE 490gz - Lecture 7 - Winter 2004

20

LZW Decoding Example (6b)

Dictionary

0	a	012436
1	b	a b ab aba ba bab
2	ab	
3	ba	
4	aba	
5	abab	
6	bab	
7	bab?	

CSE 490gz - Lecture 7 - Winter 2004

21

Decoding Exercise

Base Dictionary

0	a	0 1 4 0 2 0 3 5 7
1	b	
2	c	
3	d	
4	r	

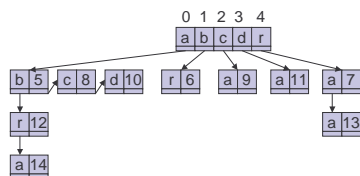
CSE 490gz - Lecture 7 - Winter 2004

22

Trie Data Structure for Encoder's Dictionary

- Fredkin (1960)

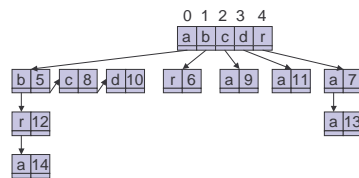
0	a	9	ca
1	b	10	ad
2	c	11	da
3	d	12	abr
4	r	13	raa
5	ab	14	abra
6	br		
7	ra		
8	ac		



CSE 490gz - Lecture 7 - Winter 2004

23

Encoder Uses a Trie (1)

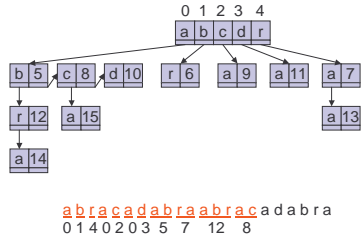


abraacadabraabraacadabra
0 1 4 0 2 0 3 5 7 12

CSE 490gz - Lecture 7 - Winter 2004

24

Encoder Uses a Trie (2)



CSE 490gz - Lecture 7 - Winter 2004

25

Decoder's Data Structure

- Simply an array of strings

0	a	9	ca
1	b	10	ad
2	c	11	da
3	d	12	abr
4	r	13	raa
5	ab	14	abr?
6	br		
7	ra		
8	ac		

0 1 4 0 2 0 3 5 7 12 8 ...
a b r a c a d a b r a a b r

CSE 490gz - Lecture 7 - Winter 2004

26

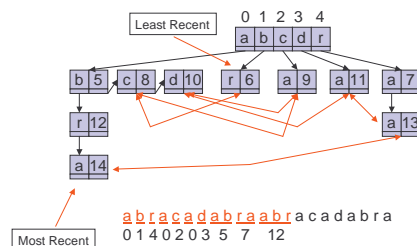
Bounded Size Dictionary

- Bounded Size Dictionary
 - n bits of index allows a dictionary of size 2^n
 - Doubtful that long entries in the dictionary will be useful.
- Strategies when the dictionary reaches its limit.
 - Don't add more, just use what is there.
 - Throw it away and start a new dictionary.
 - Double the dictionary, adding one more bit to indices.
 - Throw out the least recently visited entry to make room for the new entry.

CSE 490gz - Lecture 7 - Winter 2004

27

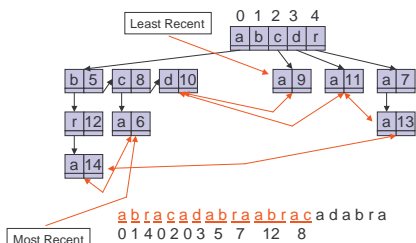
Implementing the LRV Strategy



CSE 490gz - Lecture 7 - Winter 2004

28

Implementing the LRV Strategy



CSE 490gz - Lecture 7 - Winter 2004

29

Notes on LZW

- Extremely effective when there are repeated patterns in the data that are widely spread.
- Negative: Creates entries in the dictionary that may never be used.
- Applications:
 - Unix compress, GIF, V.42 bis modem standard

CSE 490gz - Lecture 7 - Winter 2004

30