CSE 490 G
Introduction to Data Compression
Winter 2006

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

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
```

LZW Encoding Example (1)

Dictionary

0 a
1 b

a b a b a b a b a

LZW Encoding Example (2)

Dictionary

0 a
1 b
2 ab

a b a b a b a b a
0

LZW Encoding Example (3)

Dictionary

0 a
1 b
2 ab
3 ba

a b a b a b a b a
0 1

LZW Encoding Example (4)

Dictionary

0 a
1 b
2 ab
3 ba
4 aba

a b a b a b a b a
0 1 2

LZW Encoding Example (5)

Dictionary

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

a b a b a b a b a
0 1 2 4

LZW Encoding Example (6)

Dictionary

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

a b a b a b a b a
0 1 2 4 3

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 a
1 b
2 a?

0 1 2 4 3 6
a

LZW Decoding Example (2a)

Dictionary

0 a
1 b
2 ab

0 1 2 4 3 6
a b

LZW Decoding Example (2b)

Dictionary

0 a
1 b
2 ab
3 b?

0 1 2 4 3 6
a b

LZW Decoding Example (3a)

Dictionary

0 a
1 b
2 ab
3 ba

0 1 2 4 3 6
a b a

LZW Decoding Example (3b)

Dictionary

0 a
1 b
2 ab
3 ba
4 ab?

0 1 2 4 3 6
a b ab

LZW Decoding Example (4a)

Dictionary

0 a
1 b
2 ab
3 ba
4 aba

0 1 2 4 3 6
a b ab a

LZW Decoding Example (4b)

Dictionary

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

0 1 2 4 3 6
a b ab aba

LZW Decoding Example (5a)

Dictionary

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

0 1 2 4 3 6
a b ab aba b

LZW Decoding Example (5b)

Dictionary

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

0 1 2 4 3 6
a b ab aba ba

LZW Decoding Example (6a)

Dictionary

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

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

LZW Decoding Example (6b)

Dictionary

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

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

Decoding Exercise

Base Dictionary

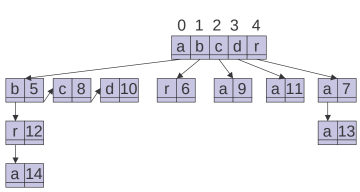
0 a
1 b
2 c
3 d
4 r

0 1 4 0 2 0 3 5 7

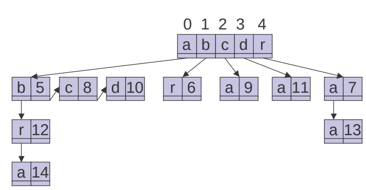
Trie Data Structure for Encoder's Dictionary

- 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		

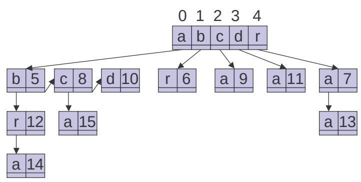


Encoder Uses a Trie (1)



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

Encoder Uses a Trie (2)



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

Decoder's Data Structure

- Simply an array of strings

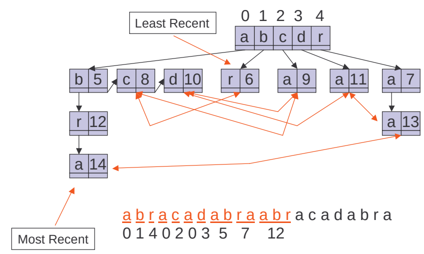
0	a	9	ca
1	b	10	ad
2	c	11	da
3	d	12	abr
4	r	13	raa
5	ab	14	abr?
6	br		
7	ra		
8	ac		

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

Bounded Size Dictionary

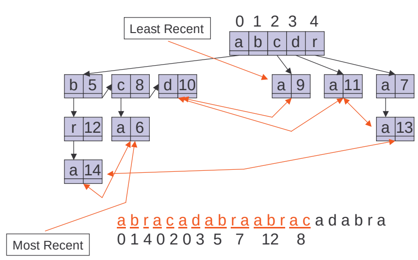
- Bounded Size Dictionary
  - n bits of index allows a dictionary of size 2^n
  - Doubtful that long entries in the dictionary will be useful.
- Strategies when the dictionary reaches its limit.
  1. Don't add more, just use what is there.
  2. Throw it away and start a new dictionary.
  3. Double the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

Implementing the LRV Strategy



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

Implementing the LRV Strategy



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

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