

# CS 199 Lectures 5-6

Data Representation:  
Text

# Text

- How many symbols can we represent using  $n$  binary digits?
- How many text symbols do we need to represent?
- 7-digit ASCII, EBCDIC

# ASCII

- American Standard Code for Information Interchange
- A-Z = 65-90                   (65=100 0001)
- a-z = 97-122                 (97=110 0001)
- 0-9 = 48-57                 (011 0000) to (011 1001)
- What about that 8th bit (to get a *byte*) ??
- *parity check*

# Extending ASCII: Unicode

- With more bits, we can represent more symbols.
- Hangul, Zhuyin, Devanagari,
- Mongolian, Cherokee, Canadian Aboriginal Syllabics, Tifinagh (Berbers), Osmanya (Somali)
- Ogham (Irish), Cuneiform, Klingon, Tolkien

# Text Compression

- several techniques
  - codeword/table lookup for common words
    - [court reporters, shorthand notation...]
  - prefix<sub>[-free]</sub> codes, Huffman coding
    - ([www.mathmaniacs.org/lessons/02-textcomp](http://www.mathmaniacs.org/lessons/02-textcomp))
  - Run-length coding, Lempel-Ziv-Welch



# Lempel- Ziv-Welch

The compressor algorithm builds a string translation table from the text being compressed. The string translation table maps fixed-length codes (usually 12-bit) to strings.

The string table is initialized with all single-character strings (256 entries in the case of 8-bit characters).

As the compressor character-serially examines the text, it stores every unique two-character string into the table as a code/character concatenation, with the code mapping to the corresponding first character.

As each two-character string is stored, the first character is outputted.

Whenever a previously-encountered string is read from the input, the longest such previously-encountered string is determined, and then the code for this string concatenated with the extension character (the next character in the input) is stored in the table.

The code for this longest previously-encountered string is outputted and the extension character is used as the beginning of the next string.

*[http:// en.wikipedia.org/wiki/LZW](http://en.wikipedia.org/wiki/LZW)*

# Lempel- Ziv-Welch

The decompressor algorithm only requires the compressed text as an input, since it can build an identical string table from the compressed text as it is recreating the original text.

However, an abnormal case shows up whenever the sequence *character/string/character/string/character* (with the same character for each *character* and string for each *string*) is encountered in the input and *character/string* is already stored in the string table. When the decompressor reads the code for *character/string/character* in the input, it cannot resolve it because it has not yet stored this code in its table. This special case can be dealt with because the decompressor knows that the extension character is the previously-encountered *character*.[\[1\]](#)