Codificação baseada em dicionário

- A codificação baseada em dicionário mais simples, é codificar cada palavra apenas com o número da página do dicionário e a ordem da palavra na folha.
  - Codificação estática
  - Desvantagem: o dicionário tem de ser transmitido

- Dicionários
  - Estáticos
  - Dinâmicos

- Vantagem: Não precisa de conhecer a estatística dos dados.

- Como funciona:
  - Substitui sequências de símbolos por referências ao dicionário.

Codificação baseada em dicionário

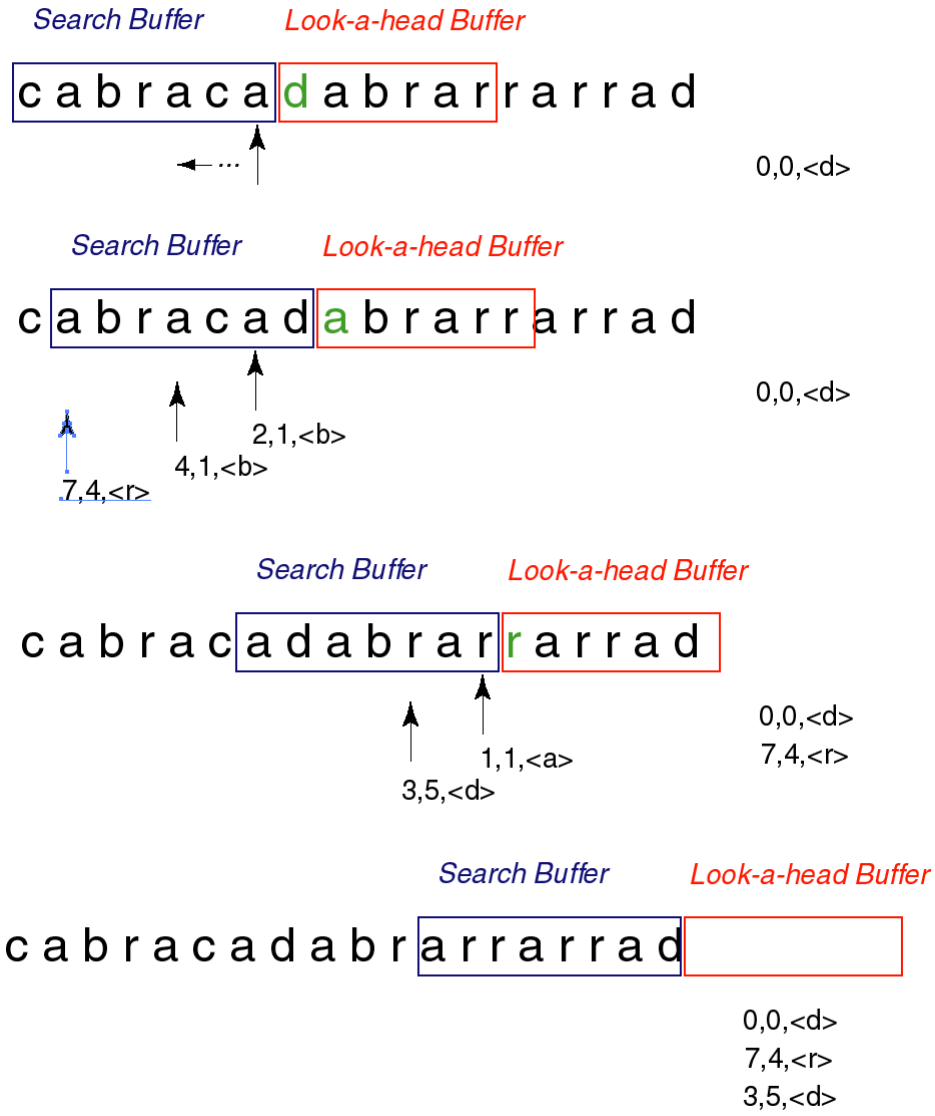
- Algoritmo LZ77
  - A sequência de símbolos a codificar é substituída por uma referência a ocorrências anteriores;
  - O dicionário é construído no compressor e no descompressor;
  - Código tem comprimento constante.

- Como funciona:
  - O codificador faz deslizar uma janela com duas partes S*earch Buffer* e o *Look-a-head Buffer* (ambos de tamanho fixo);
  - O objectivo é codificar o *Look-a-head Buffer;*
  - O dicionário é o *Search Buffer* (porção dos dados já codificada)

  - Deslocar o *Search Pointer* para encontrar um símbolo igual ao primeiro símbolo do *Look-a-head Buffer;*
  - Contar o número de símbolos iguais nos dois buffers;
  - Gerar o trio <Offset, Length Match, Code(next simbol)>
    cujo o *Length Match* é maior;
  - Deslocar a janela.

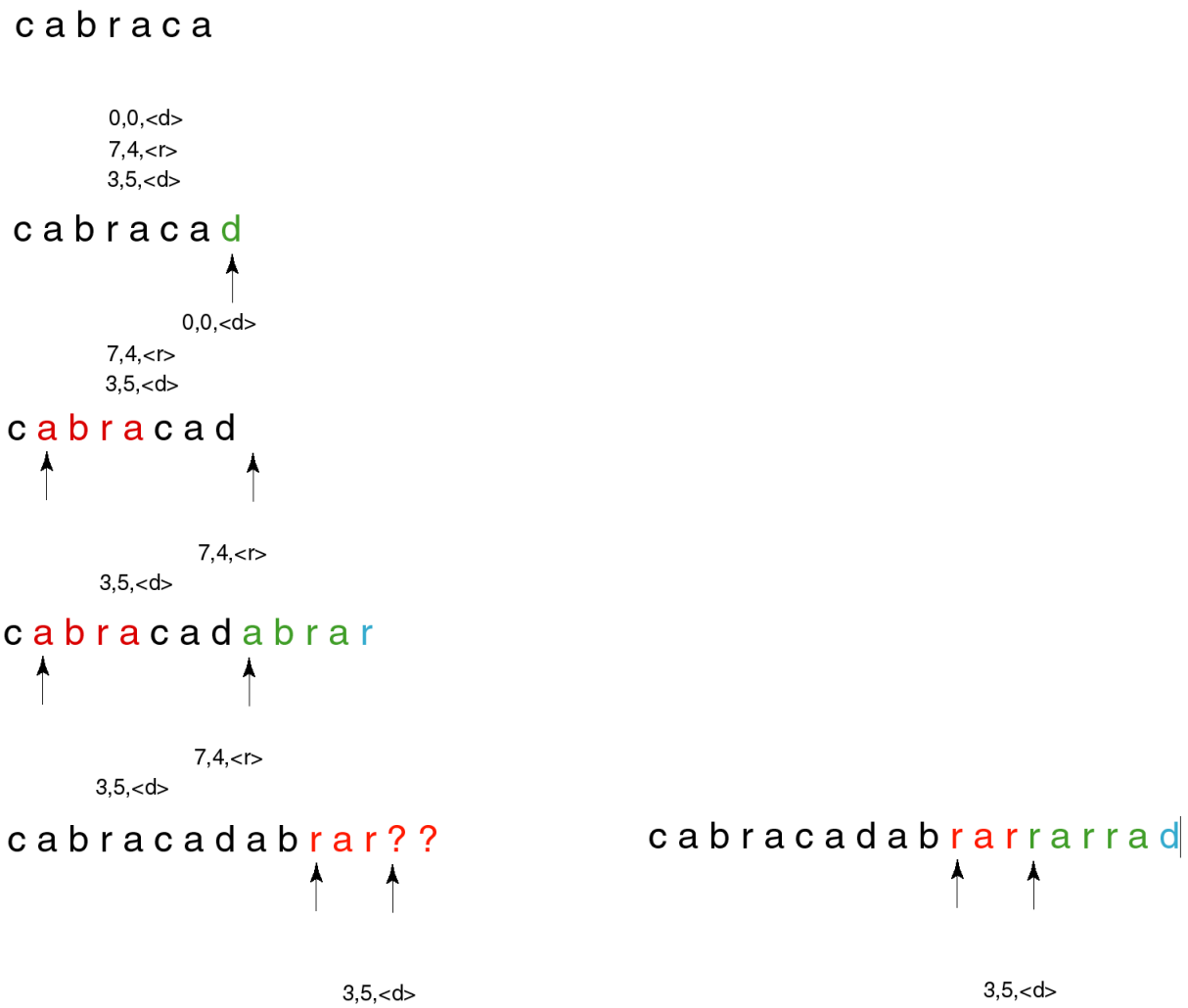
Algoritmo de Compressão LZ77

- Supondo que já se tinha codificado o *Search Buffer* na seguinte situação:



Algoritmo de Descompressão LZ77

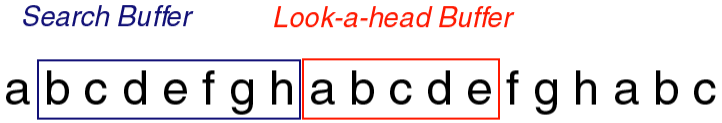
- Supondo que a dado momento já se tinha descodificado o seguinte:



Algoritmo LZ77: Variantes

- Optimização dos trios <offset, length match, code(next simbol)> (codificar com tamanho fixo ou variável) ;
- Usar flag's para evitar a situação <0,0,code>;
- Tamanho dos buffers ser adaptativo;

- Desvantagem:



Codificação baseada em dicionário

- Algoritmo Lempel-Ziv-Welch (LZW)
  - Descende do LZ78
  - Compressão adaptativa;
  - Precisa de um dicionário inicial;
  - Dicionário é construído no codificador e no descodificador;
  - O código de cada símbolo é constante
  - Cada código representa streams (de símbolos) com comprimento variável
  - Usado nos formato GIF

Algoritmo de compressão LZW

```
w = read simbol
while ( not end)
        s = read simbol
        if  [w s] exists in dictionary
                w  = [w s]
        else
              write code for w
              add string [w s] to dictionary with a new code
              w = s
        end
end
write code for s
```

- O LZW original tinha um dicionário com 4096 entradas cujas primeiras 256 eram o código ASCII

Algoritmo Lempel-Ziv-Welch

- mensagem: “ABABBABCABBABBAC”
- Assumindo apenas carateres ASCII
  - 16 simbolos x 8bits = 128 bits
- Código enviado: “1 2 4 5 2 3 6 10 3”
- Dado que se excedeu os 256 símbolos da tabela ASCII, tenho de enviar mais que 8 bits por cada símbolo

0	0	0	0	x	x	x	x	x	x	x	x

- Com 12 bit o dicionário pode ter 4096 entradas.
- Assim envio 9 códigos x 12 bit = 108 bits.

			Dictionary	
string **w**	symbol **s**	**output**	string	code
			A	1
			B	2
			C	3
A	B	1	AB	4
B	A	2	BA	5
A	B			
AB	B	4	ABB	6
B	A			
BA	B	5	BAB	7
B	C	2	BC	8
C	A	3	CA	9
A	B			
AB	B			
ABB	A	6	ABBA	10
A	B			
AB	B			
ABB	A			
ABBA	C	10	ABBAC	11
C		3		