

# Introduction to Information Theory

## DSA I 325

*Prepared By*  
**Dr. Khaled Mostafa ElSayed**  
[kmelsayed@zewailcity.edu.eg](mailto:kmelsayed@zewailcity.edu.eg)

**Instructor:**

**Dr. Khaled Mostafa ElSayed**  
[kmelsayed@zewailcity.edu.eg](mailto:kmelsayed@zewailcity.edu.eg)



**جامعة العلوم والتكنولوجيا**  
University of Science and Technology

# Introduction to Compression Techniques

# Course Reference

**“Introduction to Data Compression”**

By Khalid Sayood

*Latest Edition*

(The Morgan Kaufmann Series in Multimedia Information and Systems)

# Why Compress?

- To reduce the volume of data to be transmitted (text, fax, images)
- To reduce storage requirements (speech, audio, video)
- To reduce the bandwidth required for transmission

# Image Data Size

## Gray Image (one Byte / Pixel)

- For 1024\*768 Pixel Gray Image
- Original Size =  $1024 * 768 * 1$  Byte = 768 K bytes

o  $\rightarrow$  s  $\rightarrow$  8 bits

## Color Image (Three Bytes / Pixel {Red, Green, Blue})

- For 1024\*768 Pixel Color Image
- Original Size =  $1024 * 768 * 3$  Bytes = 2304 K bytes

# Video Data Size

## Video (25 Frame / Second)

For 1 Minute 1024\*768 Pixel Video clip

- Original Size (for 1 Sec) =  $1024*768 * 3 \text{ Bytes} * 25 \text{ Frames} = 57600 \text{ K bytes}$
- Original Size (for 1 Min) =  $1024*768 * 3 \text{ Bytes} * 25 \text{ Frames / Sec} * 60 \text{ Sec/Min} = 57600 * 60 = 3456000 \text{ K bytes} = 3.456 \text{ GB}$
- What About 2 Hours Movie ?? ( $3.456 * 120 \text{ Min} = !!!!$ )
- What of using NTSC system (30 Frame / Sec) !!!

# How is compression possible?

[1] Redundancy in digital audio, image, and video data

[2] Properties of human perception

Digital audio is a series of sample values;

Image is a rectangular array of pixel values;

Video is a sequence of images played out at a certain rate

***Neighboring sample values are correlated***

# Redundancy

Adjacent **audio samples** are similar (predictive encoding); samples corresponding to silence (**silence removal**)

In **digital image**, neighboring samples on a scanning line are normally similar (**spatial redundancy**)

In **digital video**, in addition to spatial redundancy, neighboring images in a video sequence may be similar (**temporal redundancy**)

# Human Perception Factors

Compressed version of digital audio, image,  
video need not represent the original information  
exactly

Perception sensitivities are different for different  
signal patterns

Human eye is less sensitive to the higher spatial  
frequency components than the lower frequencies

# Classification of Compression Techniques

## [1] Lossless compression

lossless compression for legal and medical documents, computer programs

*exploit only data redundancy*

## [2] Lossy compression

digital audio, image, video where some errors or loss can be tolerated

*exploit both data redundancy and human perception properties*

## [3] Near Lossless Compression

It is a lossy compression with a predefined max accepted error

# Classification of Compression Techniques

## [4] Hybrid Techniques

A compression algorithm that utilizes many lossy/lossless techniques to achieve high compression ratio with best quality.  
(.e.g. JPEG, MPEG, H264,..)

Constant bit rate versus variable bit rate coding ??

# Image Quality Measure

## Subjective

- Evaluated by human observers
- Do not require the original copy as a reference
- Reliable, accurate yet impractical

## Objective

- Easy to operate (automatic)
- Often requires the original copy as the reference  
(measures fidelity rather than quality)
- Works better if taking HVS model into account

# Image Quality

Gray Image 400 \* 500 Pixels

Image Size =  $400 * 500 * 1$  byte/pixel  
=200,000 byte  $\approx$  200 Kbyte

What will be the degradation in Quality if this image is compressed using lossy compression ?

- Degradation in smoothness ?
- Degradation in Eye details ?
- Degradation in Sharpness of finger edges?



# Image Quality

Gray Image 400 \* 500 Pixels

$$\begin{aligned}\text{Image Size} &= 400 * 500 * 1 \text{ byte/pixel} \\ &= 200,000 \text{ byte} \approx 200 \text{ Kbyte}\end{aligned}$$

What will be the degradation in Quality if this image is compressed using lossy compression ?

- Degradation in smoothness ?
- Degradation in Eye details ?
- Degradation in Sharpness of finger edges?



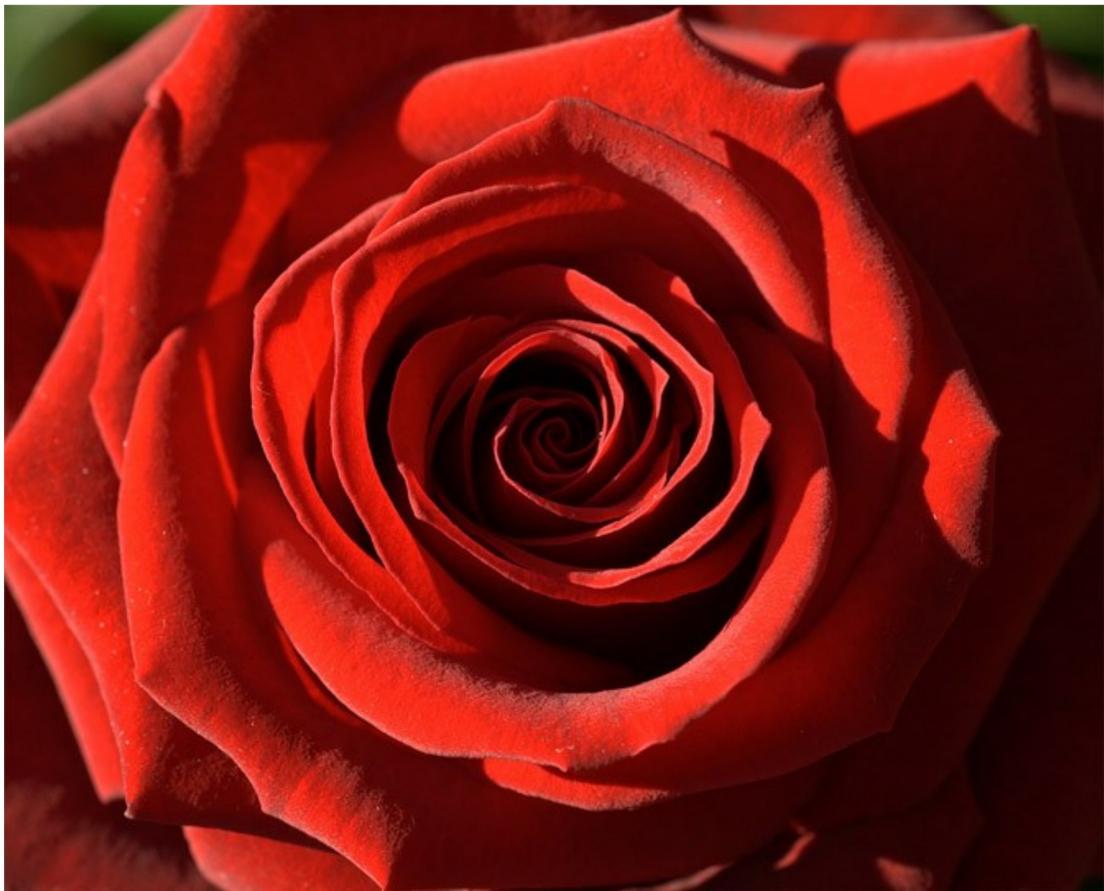
Sorry, this is the compressed version with size 38Kbyte Only  
Is this quality accepted for you ??

*The compressed size is about 1/5 of the original size*

# All These Images are Lossy Compressed images



# All These Images are Lossy Compressed images

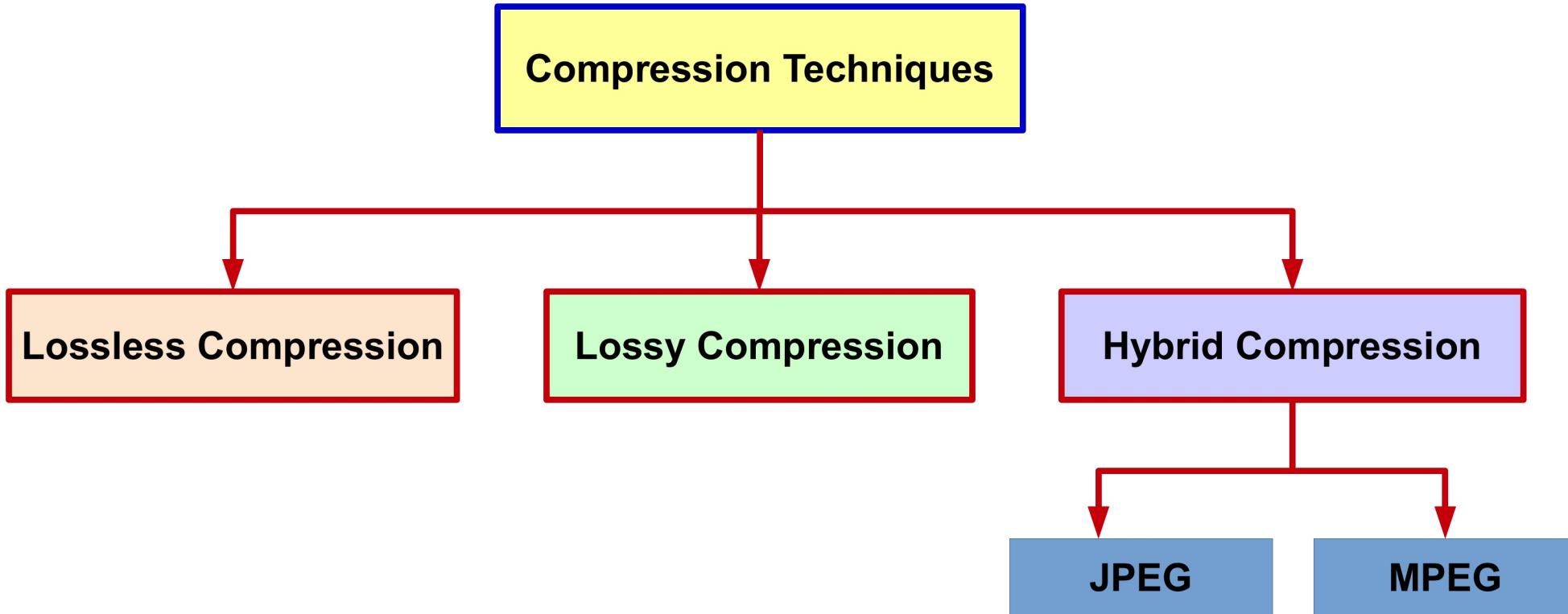


**670 \* 527 Pixels  
Original 353K  
Compressed 91K**

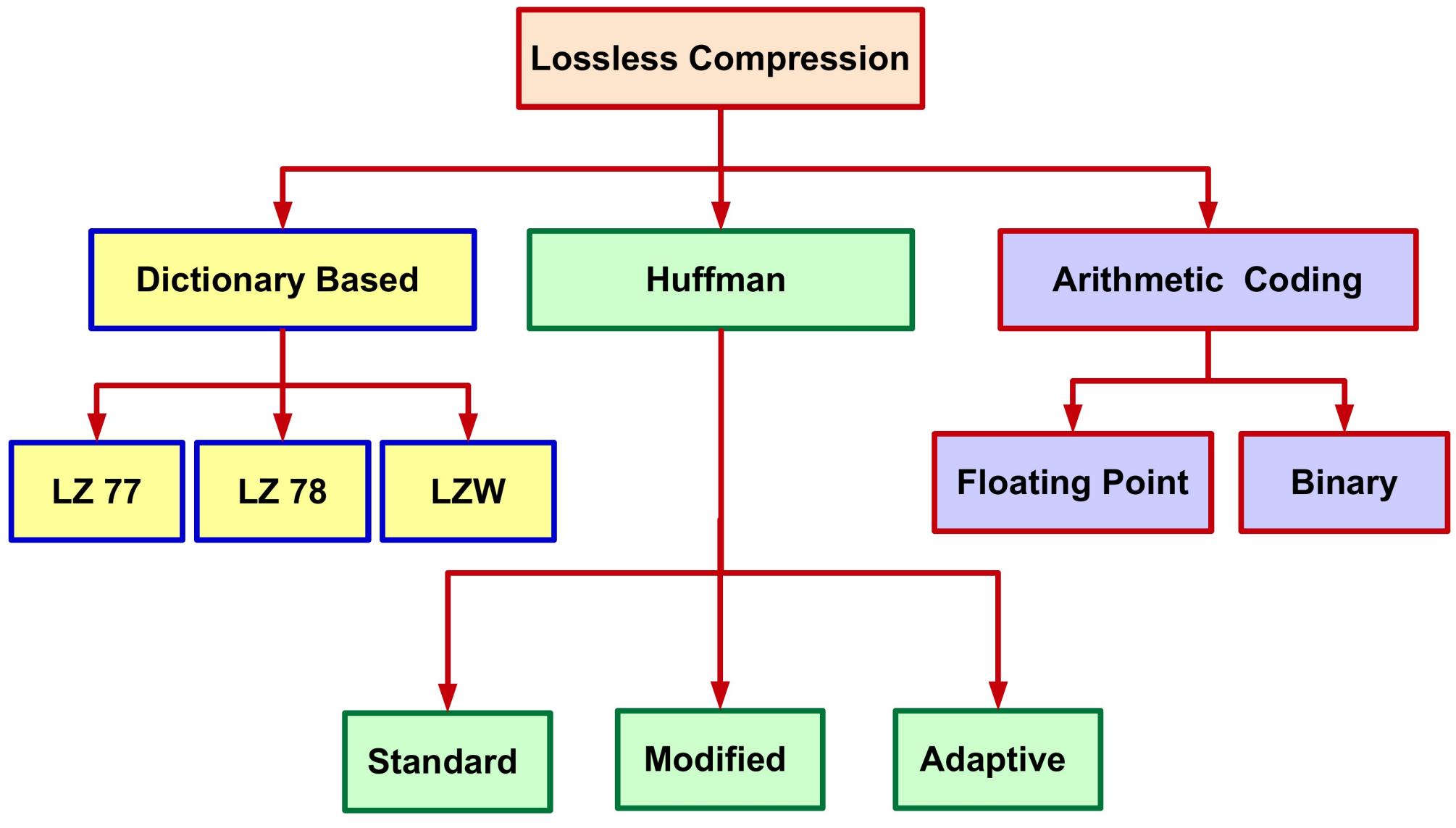


**400 \* 400 pixels  
Original 160K  
Compressed 80K**

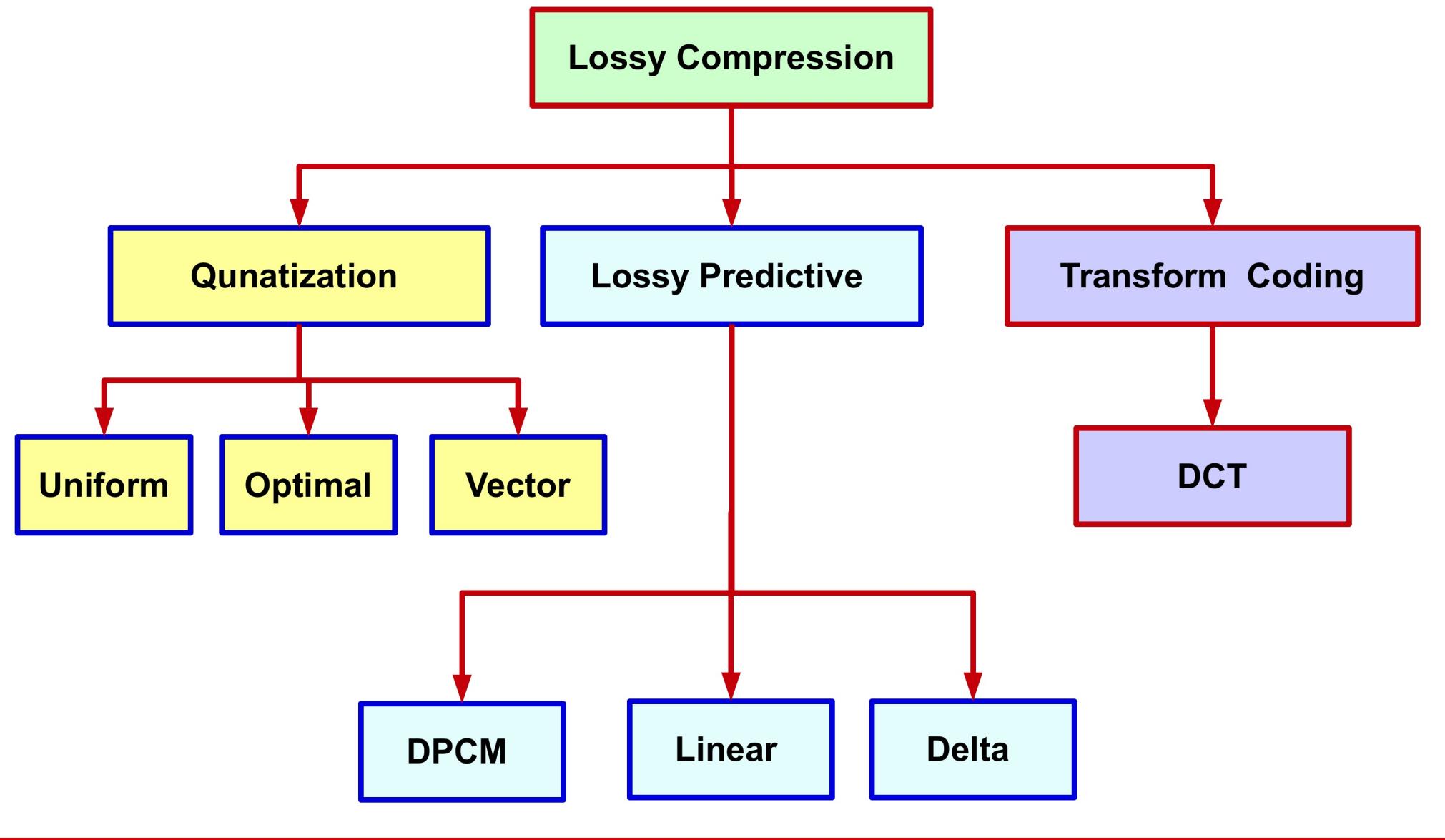
# Compression Techniques (to be covered in our course)



# Compression Techniques (to be covered in our course)



# Compression Techniques (to be covered in our course)



# Text Compression

## [1] Dictionary Based Compression

# Dictionary Based Compression

## First intuitive approach

The Students Enjoy The Math Class

The	Students	Enjoy	The	Math	Class
9	8	7	9	10	6

**Compress Code [9,11,8,11,7,11,9,11,10,11,6]**

Size of compressed Data = 11 codes \* 4 bits/code = 44 bits

Size of Original Data = 34 Char \* 8 bits / chars= 272 bits

0	Is
1	was
2	are
3	Boy
4	Girl
5	Doctor
6	Class
7	Enjoy
8	Students
9	The
10	Math
11	{Space}

# Dictionary Based Compression

## Problem: First intuitive approach

The Students Enjoy The Math Class

The      Students      Enjoy      The      Math      Class  
 9                8                7                9                10                6

**Compress Code [9,11,8,11,7,11,9,11,10,11,6]**

[1] Size of Dictionary ? (should be added to compressed file)

[2] Words not in dictionary ???

0	Is
1	was
2	are
3	Boy
4	Girl
5	Doctor
6	Class
7	Enjoy
8	Students
9	The
10	Math
11	{Space}

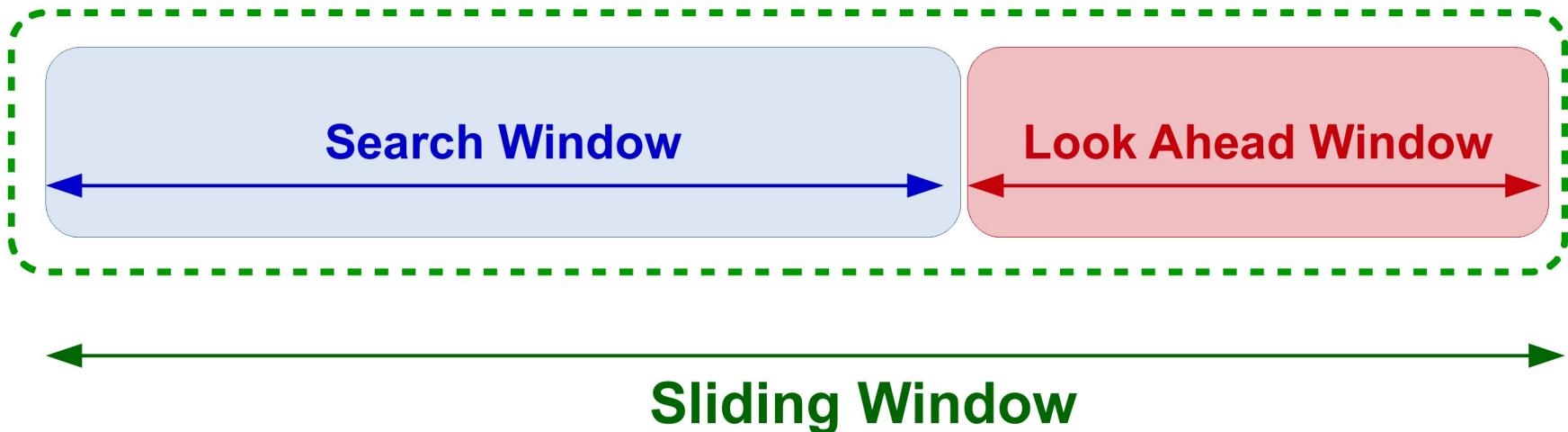
# Dictionary Based Compression

LZ 77

LZ78

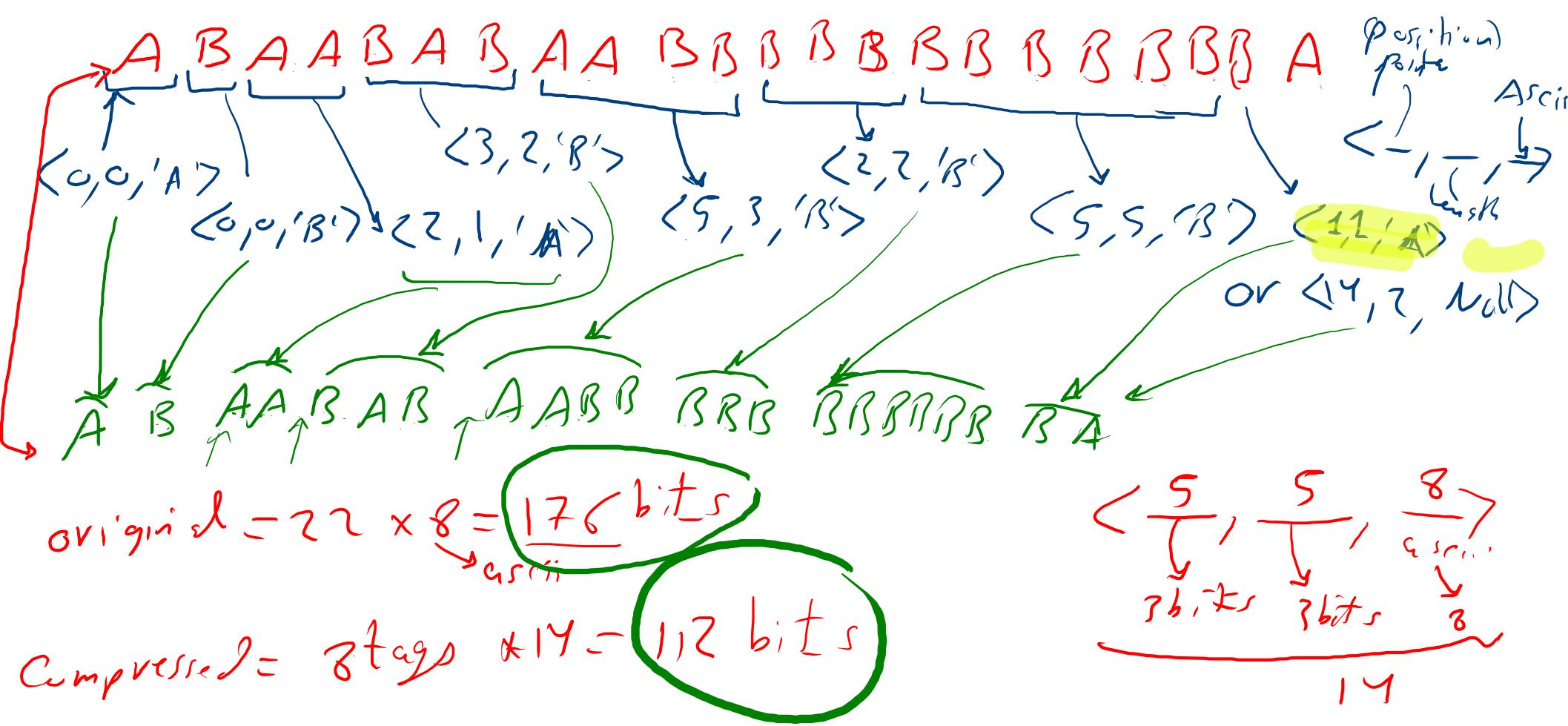
LZW

# Lempel Ziv 77 Algorithm



A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**TAG** → **<Position , Length , Next Symbol >**



# Lempel Ziv 77 Algorithm

**Search Buffer:** It contains a portion of the recently encoded sequence.

**Look-Ahead Buffer:** It contains the next portion of the sequence to be encoded.

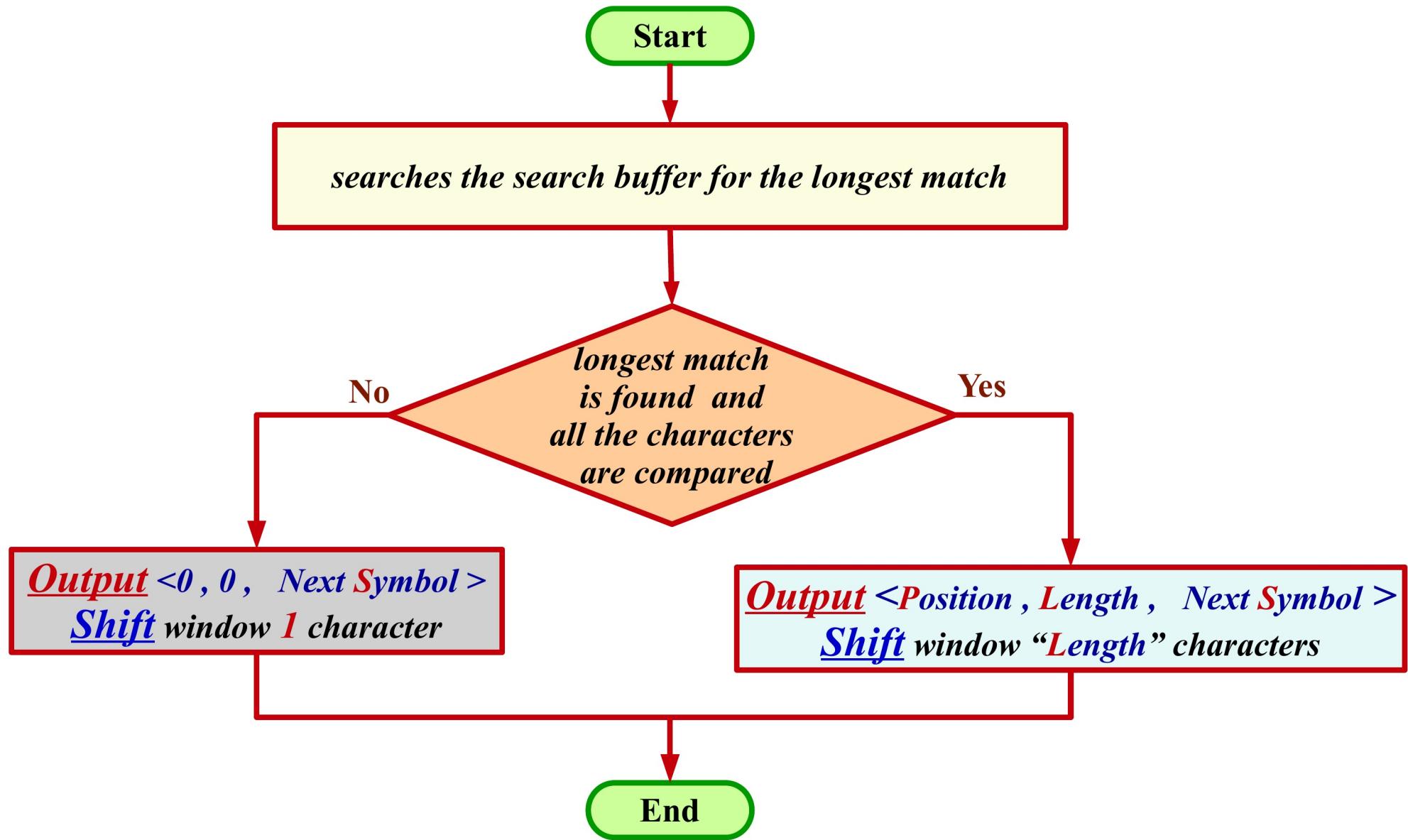
Once the longest match has been found, the encoder encodes it with a triple ***<Position , Length , Next Symbol >***

***Position*** :the offset or position of the longest match from the lookahead buffer

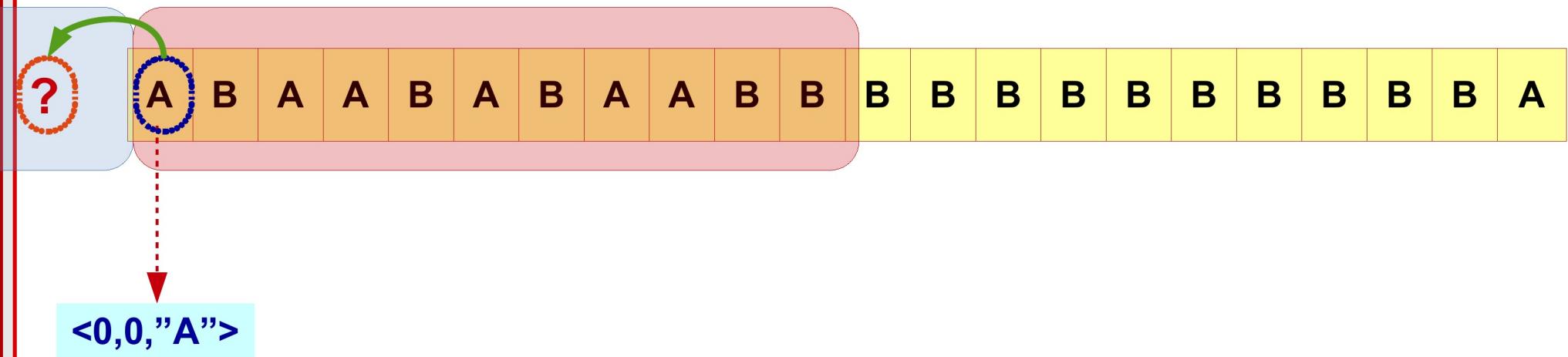
***Length*** :the length of the longest matching string

***Next Symbol*** :the codeword corresponding to the symbol in the look-ahead buffer that follows the match

# Lempel Ziv 77 Algorithm

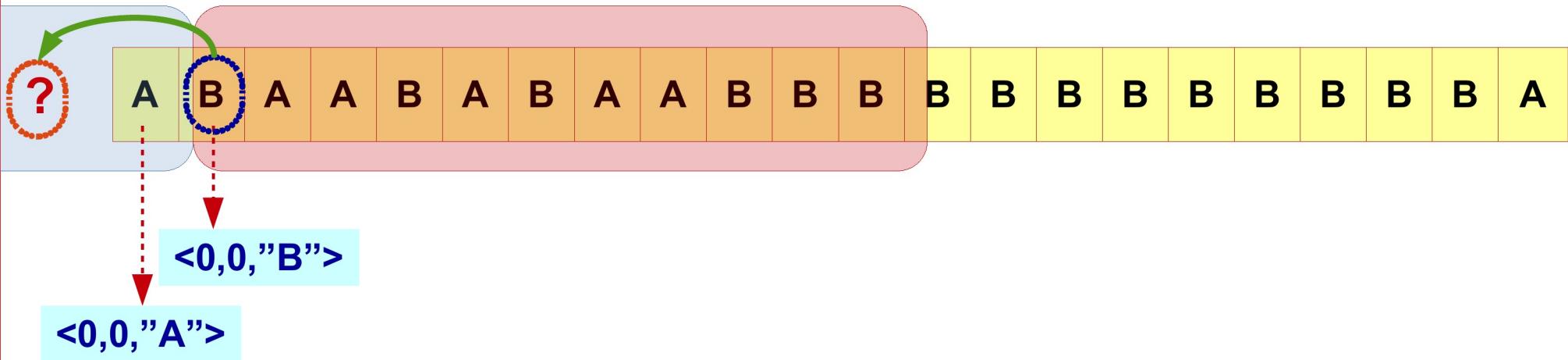


# LZ 77 (Compression)



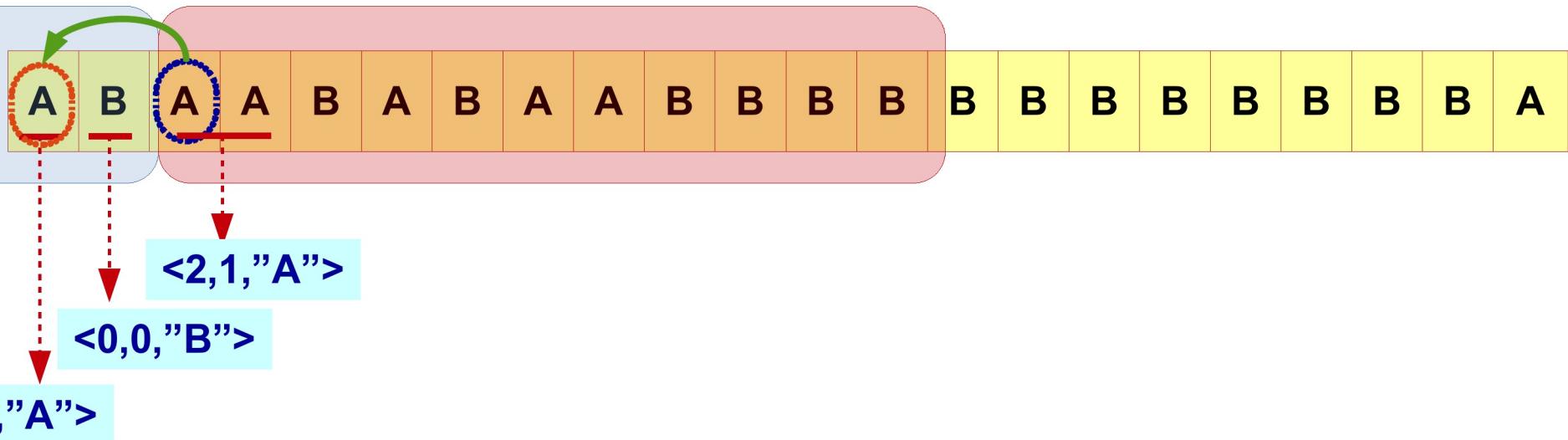
*There is no “A” in search buffer  
Position=0, Length =0, next Symbol="A"*

# LZ 77 (Compression)



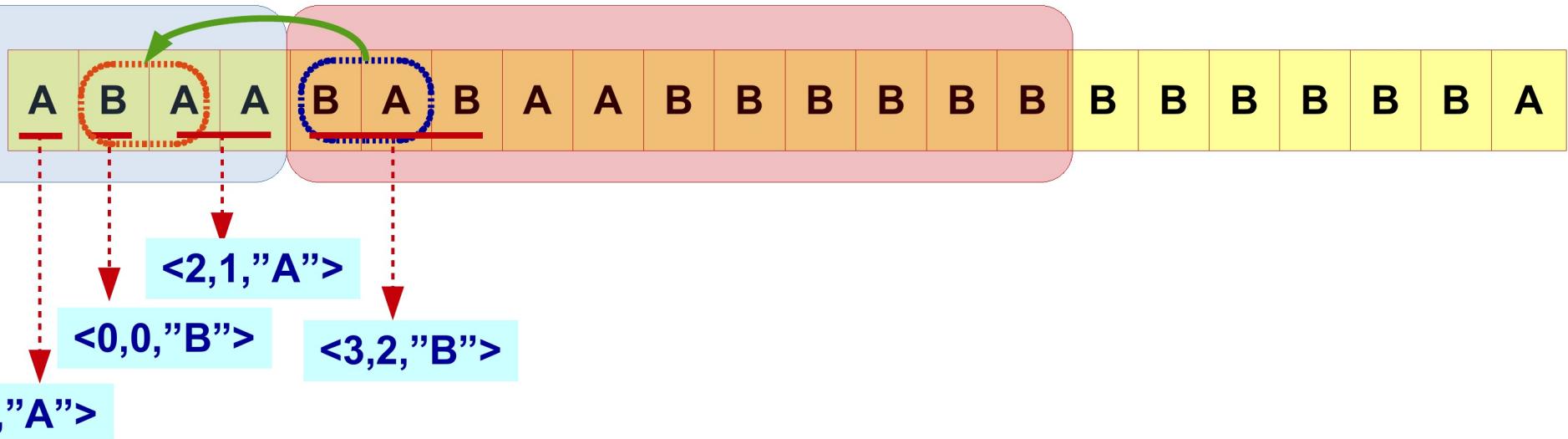
*There is no “B” in search buffer  
Position=0, Length =0, next Symbol=“B”*

# LZ 77 (Compression)



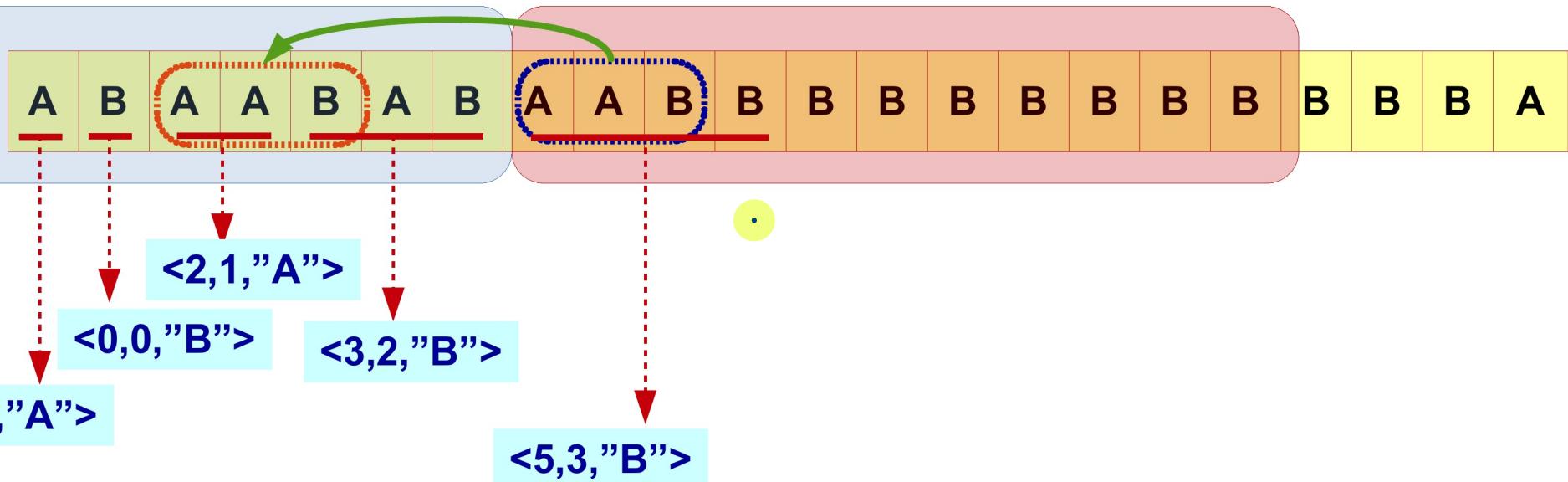
*Only “A” exists in search buffer  
 Go Back two Steps, Pick One Symbol  
 Position=2, Length =1, next Symbol=“A”*

# LZ 77 (Compression)



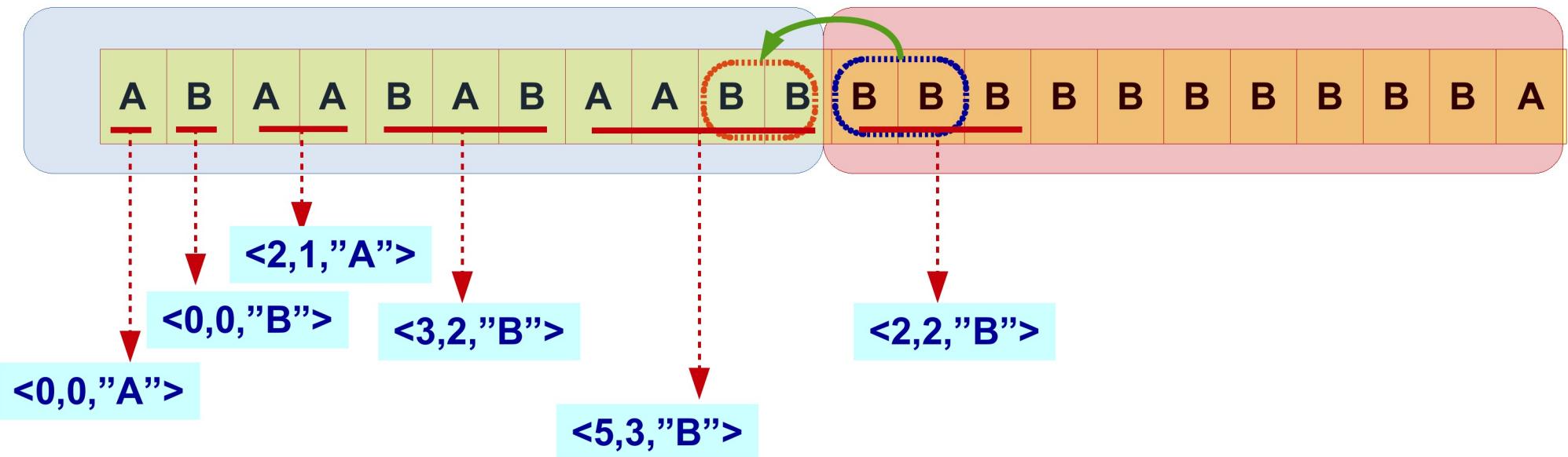
**“BA” exists in search buffer**  
**Go Back three Steps, Pick Two Symbol**  
**Position=3, Length =2, next Symbol=“B”**

# LZ 77 (Compression)



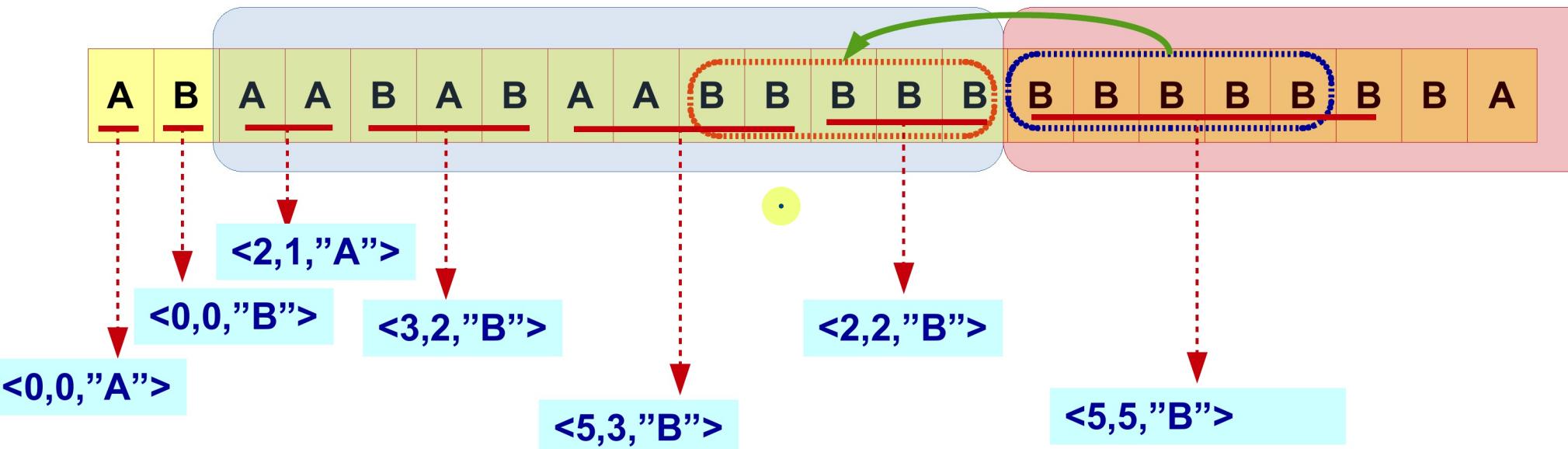
***“AAB” exists in search buffer***  
***Go Back five Steps, Pick Three Symbol***  
***Position=5, Length =3, next Symbol=“B”***

# LZ 77 (Compression)



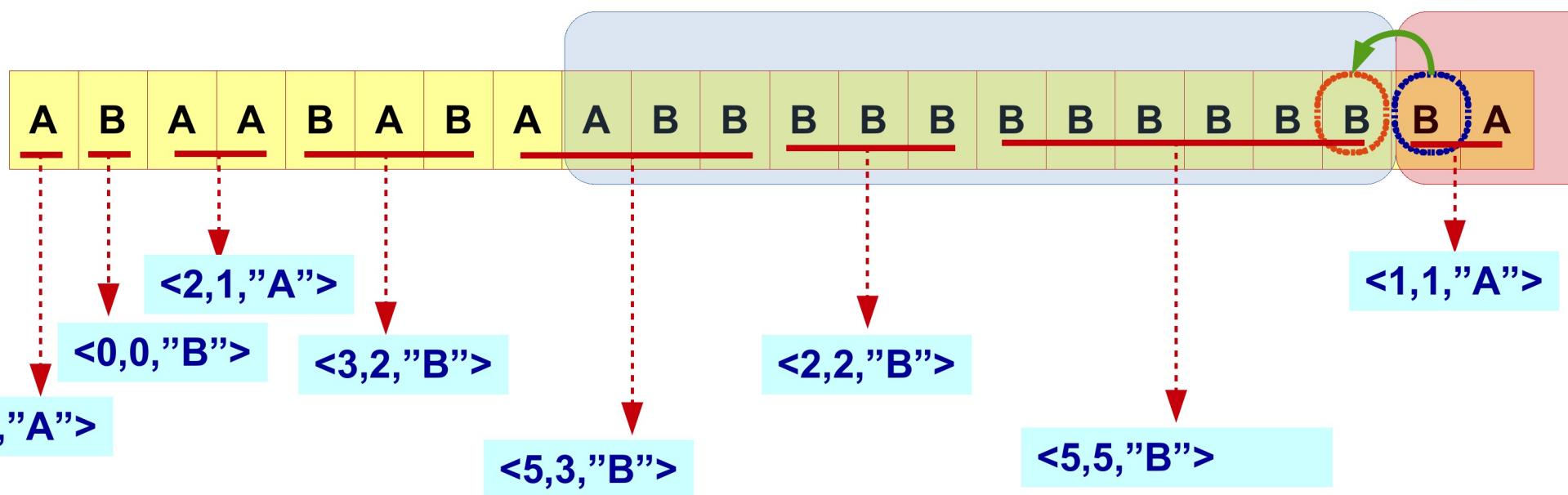
**“BB” exists in search buffer**  
**Go Back two Steps, Pick two Symbols**  
**Position=2, Length =2, next Symbol=“B”**

# LZ 77 (Compression)



**“BBBBB” exists in search buffer**  
**Go Back five Steps, Pick five Symbol**  
**Position=5, Length =5, next Symbol=“B”**

# LZ 77 (Compression)



**“B” exists in search buffer  
Go Back One Steps, Pick One Symbol  
Position=1, Length =1, next Symbol=“A”**

# LZ77 (Compression Ratio)

## Remember

**1 Bit** represent

**2 Values (0,1) [0-1]**

**2 Bits** represent

**4 values (00,01,10,11) [0-3]**

**3 Bits** represent

**8 Values (000,001,010,011,100,101,110,111) [0-7]**

## In General

**N Bits** can be used to represent  **$2^N$  Values [1 -  $2^N-1$ ]**

# LZ77 (Compression Ratio)

**Tag = < Position, Length ,Next Symbol Code>**

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">

**Original Size** = Number of Symbols \* Bits used to Store one Symbol  
= 22 Symbols \* 8 Bits / Symbol = 176 bits  
(Store “Symbol” ASCII Code in 8 Bits)

Max “Position” Value = 5

Max “Length” Value=5

Max Symbols = 256 Symbol

Tag size = 3 + 3 + 8 =14 Bits

Store “Position” Value in 3 Bits

Store “Length” Value in 3 Bits

Store “Symbol” ASCII Code in 8 Bits

Number of Tags = 8 Tags

**Compressed Size=8\*14=112 bits**

# LZ77 (Compression Ratio)

**Tag = < Position, Length ,Next Symbol Code>**

## Effect of Increasing length of Search Window

Higher Probability to find matched strings ( Decrease Number of Tags) 

Increase Number of Bits used to Store “Position” values 

## Effect of Increasing length of Look Ahead Buffer

Higher Probability to match longer strings( Decrease Number of Tags) 

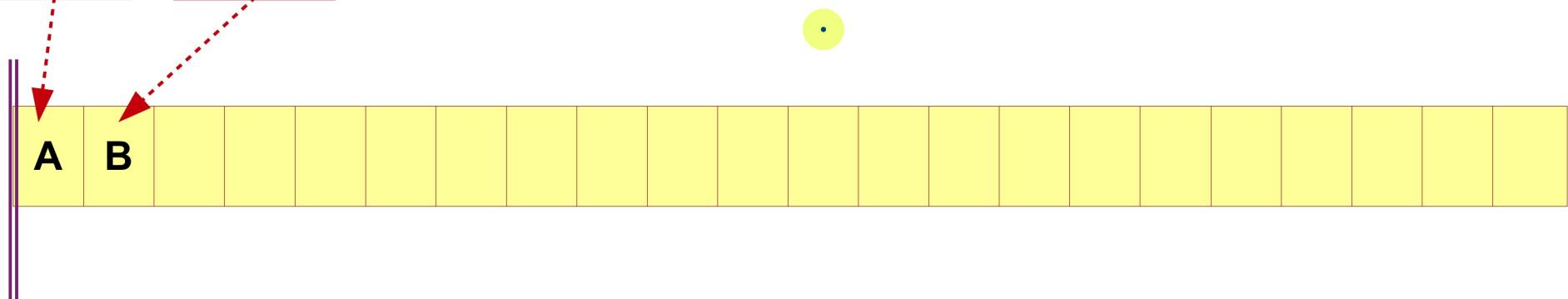
Increase Number of Bits used to Store “Length” values 

# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">



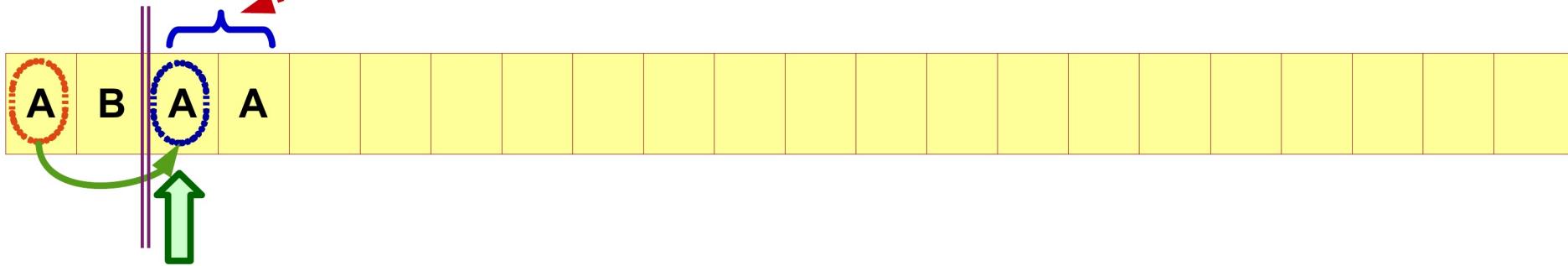
*Don't pick any symbol from Search Window*  
*Add Symbol="A"*  
*Add Symbol="B"*

# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">



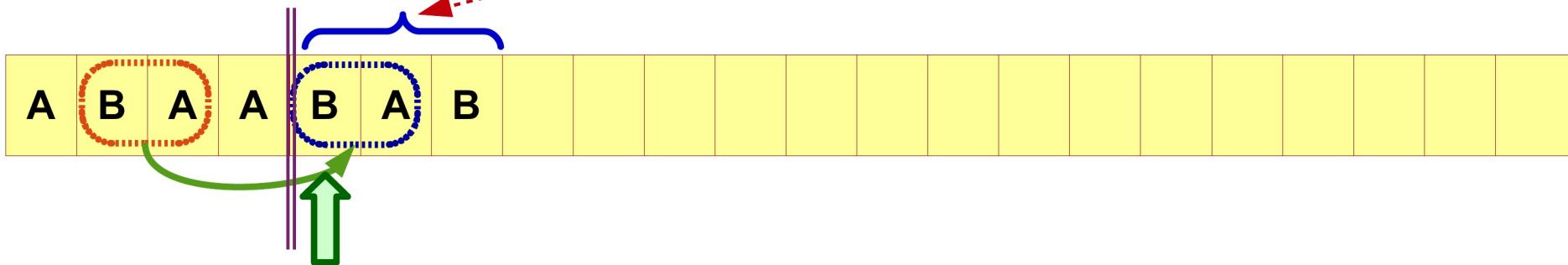
*Go back Two Positions in search window  
pick One symbol from Search Window  
Add Symbol="A"*

# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">



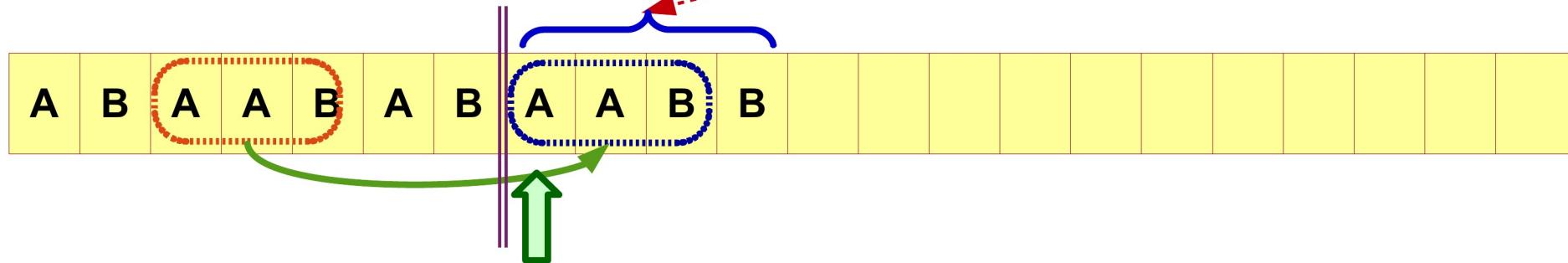
*Go back **Three Positions** in search window  
pick **Two symbols** from Search Window  
Add **Symbol="B"***

# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">



*Go back Five Positions in search window  
pick Three symbols from Search Window  
Add Symbol="B"*

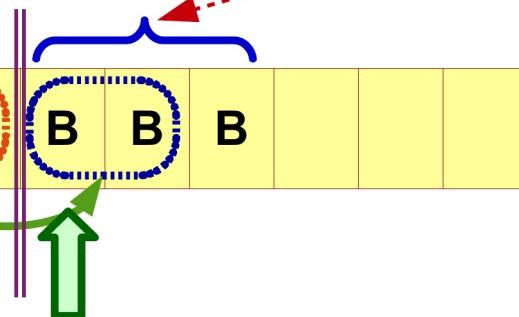
# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



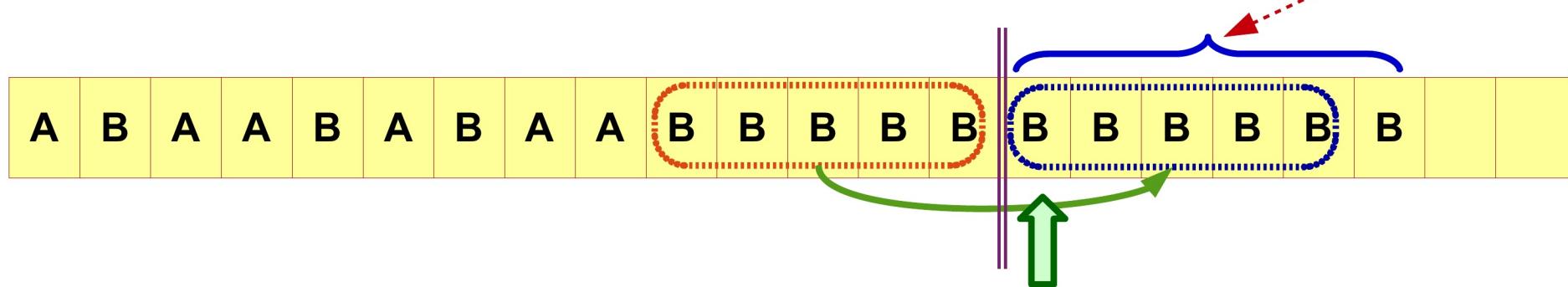
*Go back **Two Positions** in search window  
pick **Two symbols** from Search Window  
Add **Symbol="B"***

# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">



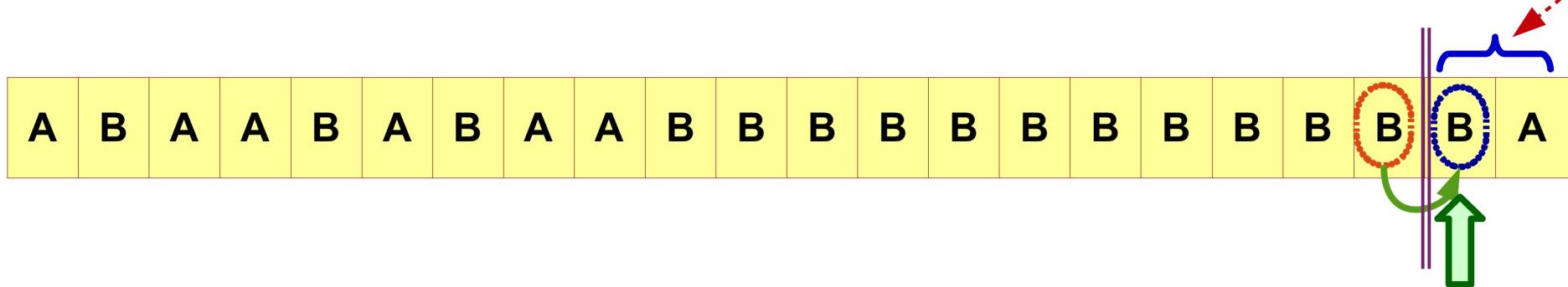
*Go back Five Positions in search window  
pick Five symbols from Search Window  
Add Symbol="B"*

# LZ 77 (Decompression)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <2,2,"B">    <5,5,"B">    <1,1,"A">



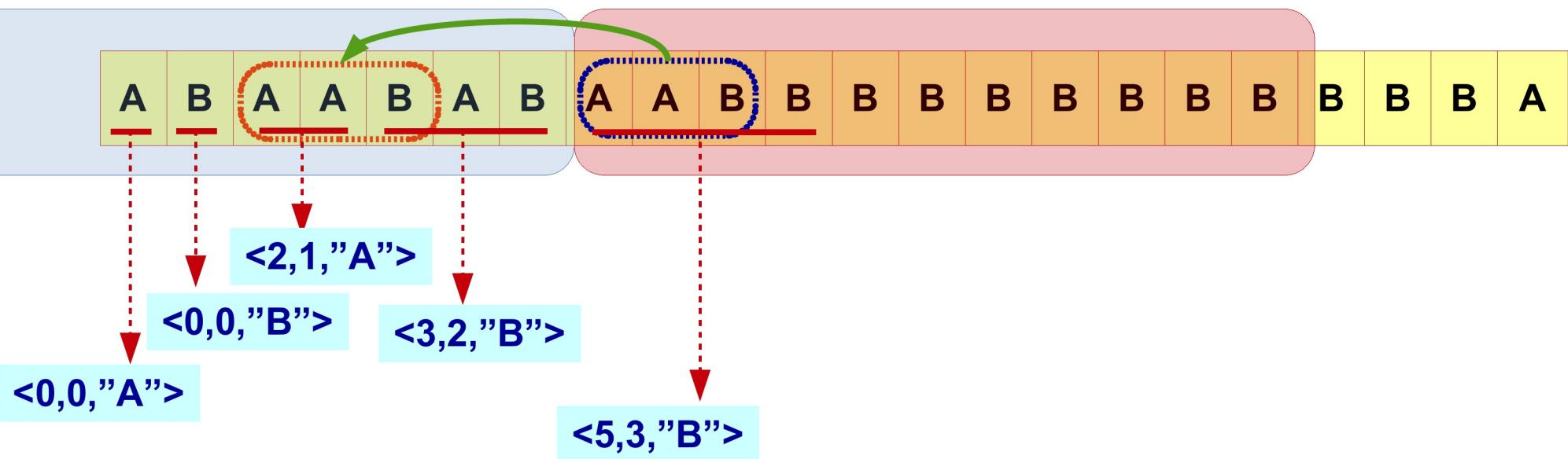
*Go back **One Positions** in search window  
pick **One symbol** from Search Window  
Add **Symbol="B"***

# LZ 77 (Compression)

## (Handling Repetitive Sequence)

*Back to Previous Example*

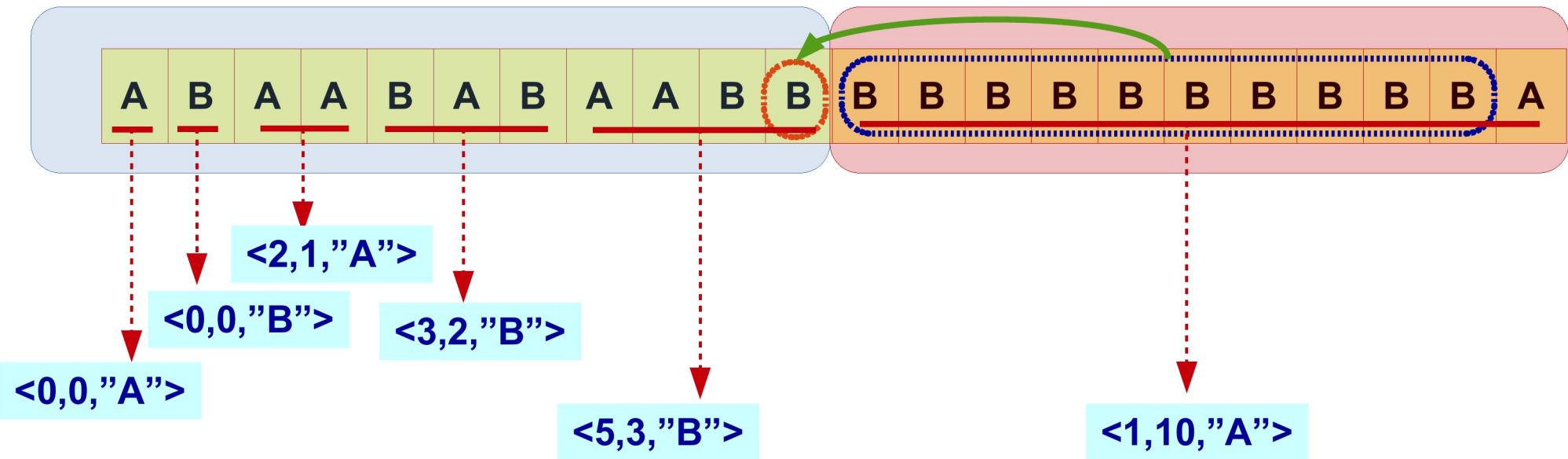
*Can We manipulate Consecutive “B”s more efficient ?*



“**AAB**” exists in search buffer  
 Go Back five Steps, Pick Three Symbol  
**Position=5, Length =3, next Symbol=“B”**

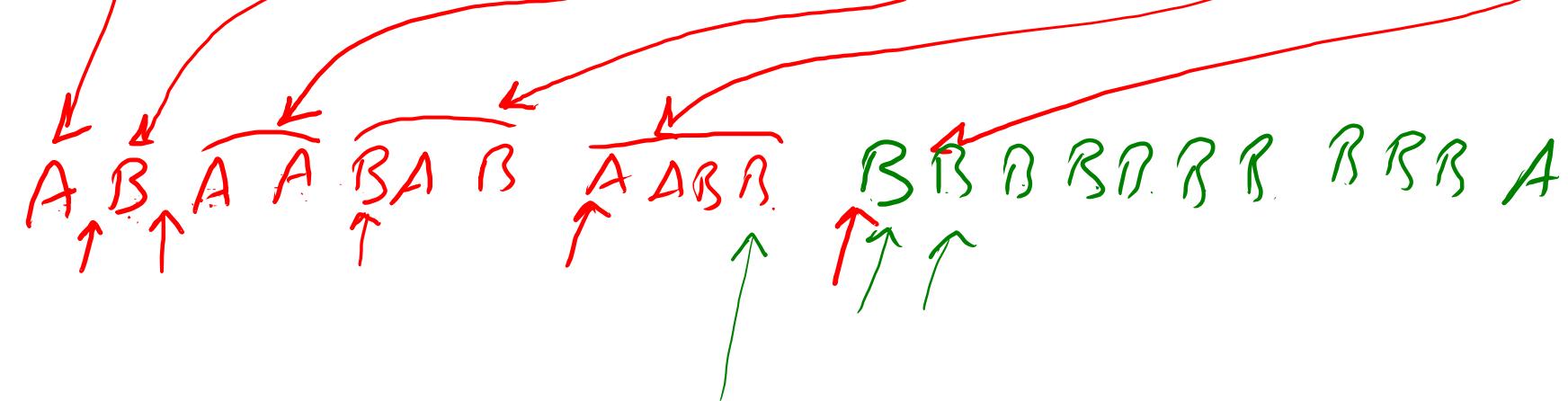
# LZ 77 (Compression) (Handling Repetitive Sequence)

*YES, We Can*



*There are Ten Consecutive "B" in Look Ahead Buffer  
"B" exists in search buffer One position Backward  
Go Back One Steps, Pick Ten Symbols  
Position=1, Length =10, next Symbol="A"*

$\langle \cdot, \cdot, A \rangle, \langle \cdot, \cdot, B \rangle, \langle 2, 1, A \rangle, \langle 3, 2, B \rangle, \langle 5, 3, B \rangle, \langle 1, 10, A \rangle$



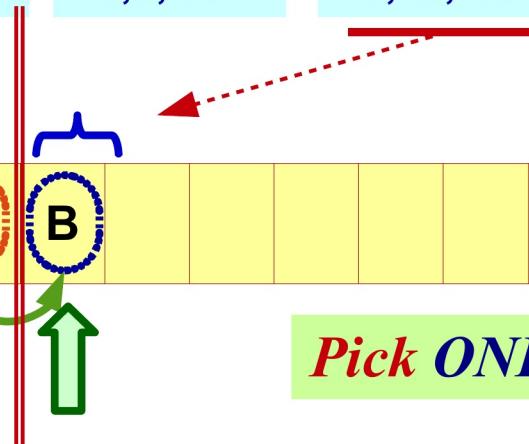
# LZ 77 (Compression) (Handling Repetitive Sequence)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <1,10,"A">

A	B	A	A	B	A	B	A	A	B	B	B										
---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--



*Go back One Position in search window  
pick Ten symbols from Search Window (in 10 Steps)  
Add Symbol="A"*

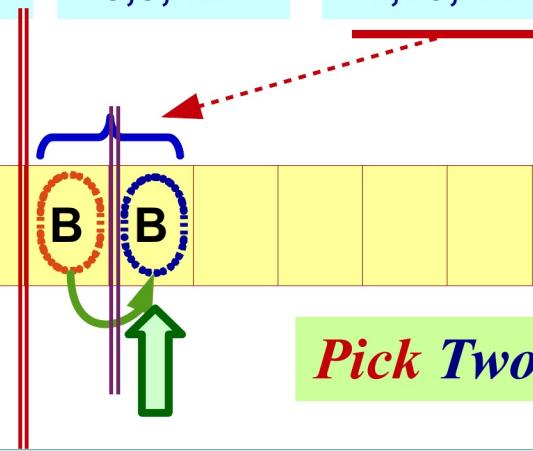
# LZ 77 (Compression) (Handling Repetitive Sequence)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <1,10,"A">

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



*Pick Two (out of Ten) Symbol*

*Go back One Position in search window  
pick Ten symbols from Search Window (in 10 Steps)  
Add Symbol="A"*

# LZ 77 (Compression) (Handling Repetitive Sequence)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <1,10,"A">

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

*Pick Three (out of Ten) Symbol*

*Go back One Position in search window  
pick Ten symbols from Search Window (in 10 Steps)  
Add Symbol="A"*

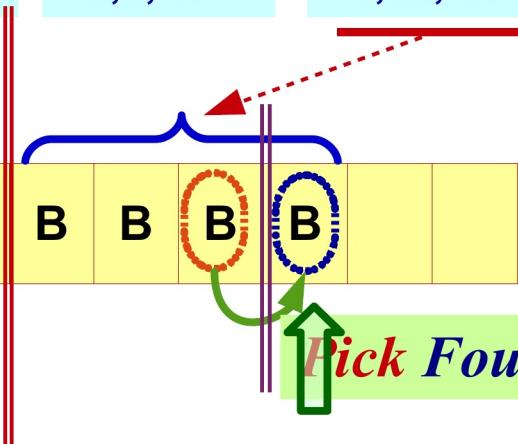
# LZ 77 (Compression) (Handling Repetitive Sequence)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <1,10,"A">

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



*Repeat , Five, Six, Seven, Eight, Nine, and Ten*

Go back **One Position** in search window  
pick **Ten symbols** from Search Window (**in 10 Steps**)  
Add **Symbol="A"**

# LZ 77 (Compression) (Handling Repetitive Sequence)

*Original Data*

A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<0,0,"A">    <0,0,"B">    <2,1,"A">    <3,2,"B">    <5,3,"B">    <1,10,"A">

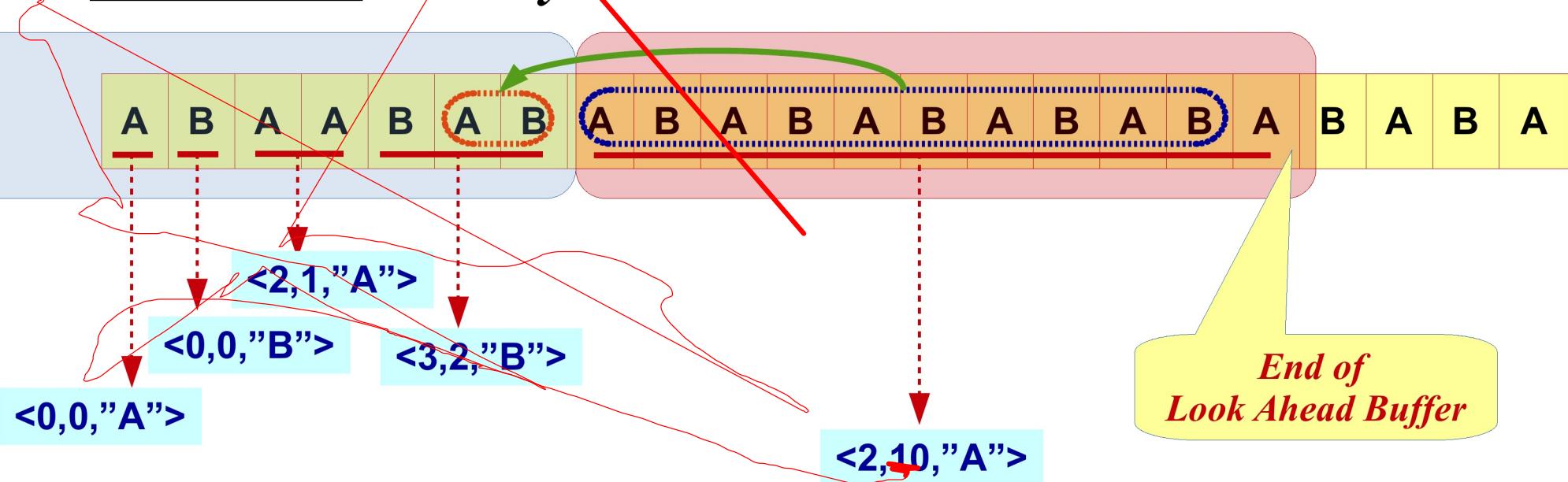
A	B	A	A	B	A	B	A	A	B	B	B	B	B	B	B	B	B	B	B	B	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

*Pick Ten (out of Ten) Symbol*

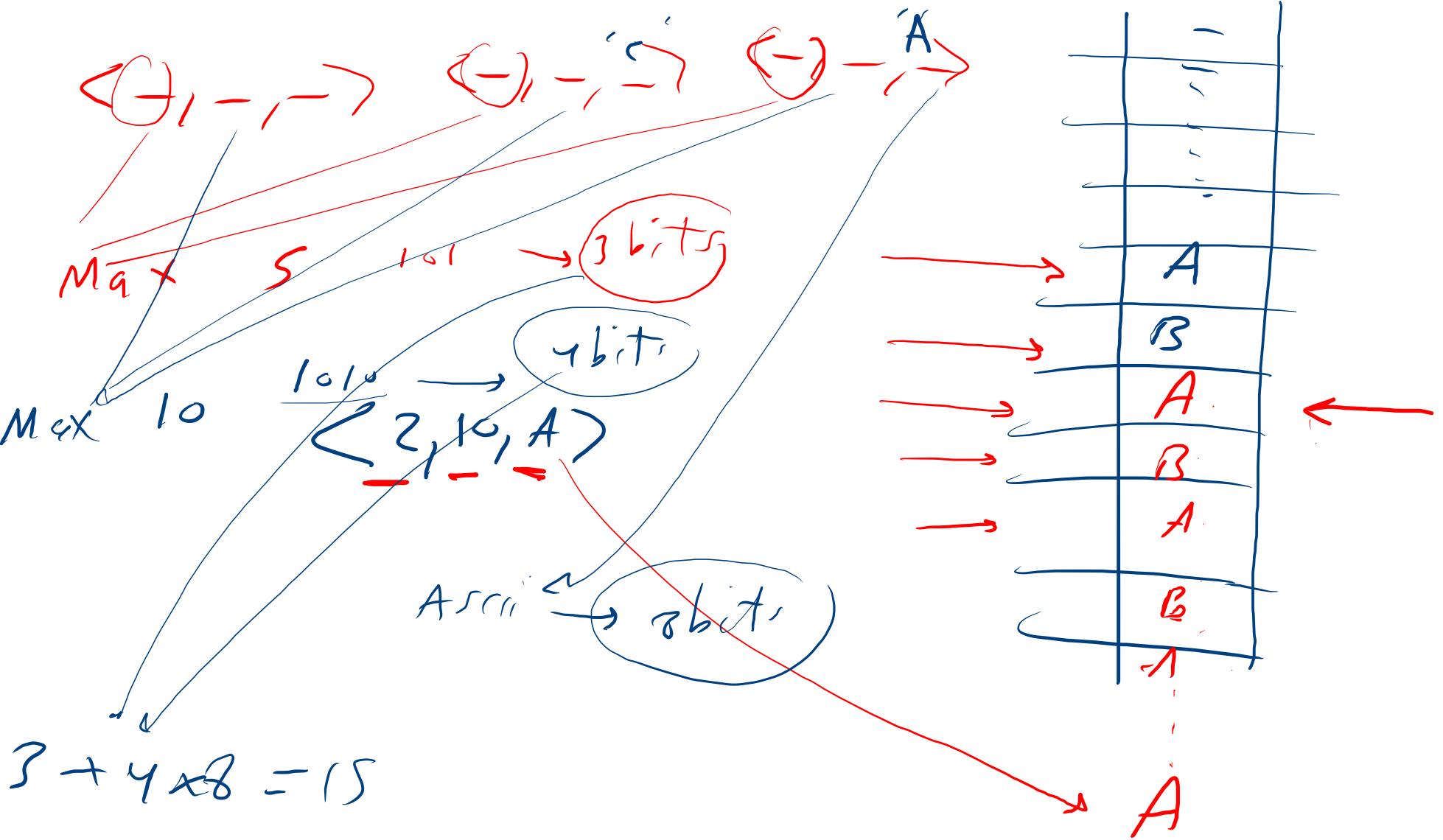
*Go back One Position in search window  
pick Ten symbols from Search Window (in 10 Steps)  
Add Symbol="A"*

# LZ 77 (Compression) (Handling Repetitive Sequence)

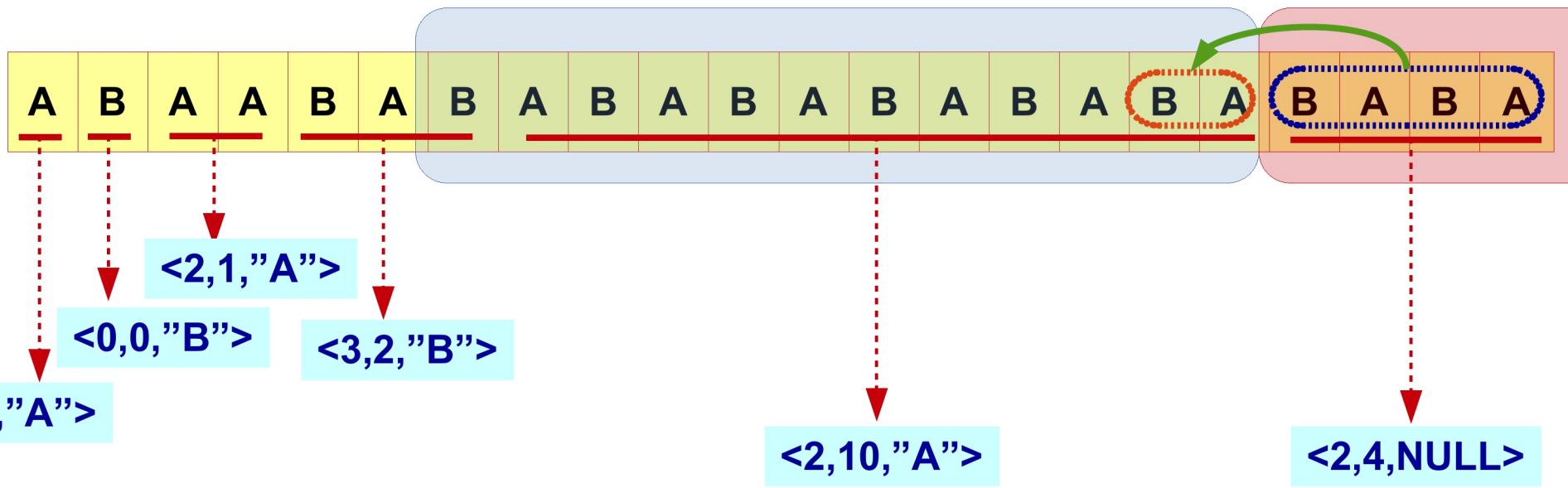
*Can We Apply the same Technique on  
Consecutive “Two Symbols” ?*



*There are Ten Consecutive Symbols “AB” in Look Ahead Buffer  
“AB” exists in search buffer Adjacent to Look Ahead Buffer  
Go Back Two Steps, Pick Ten Symbols  
Position=2, Length =10, next Symbol=“A”*



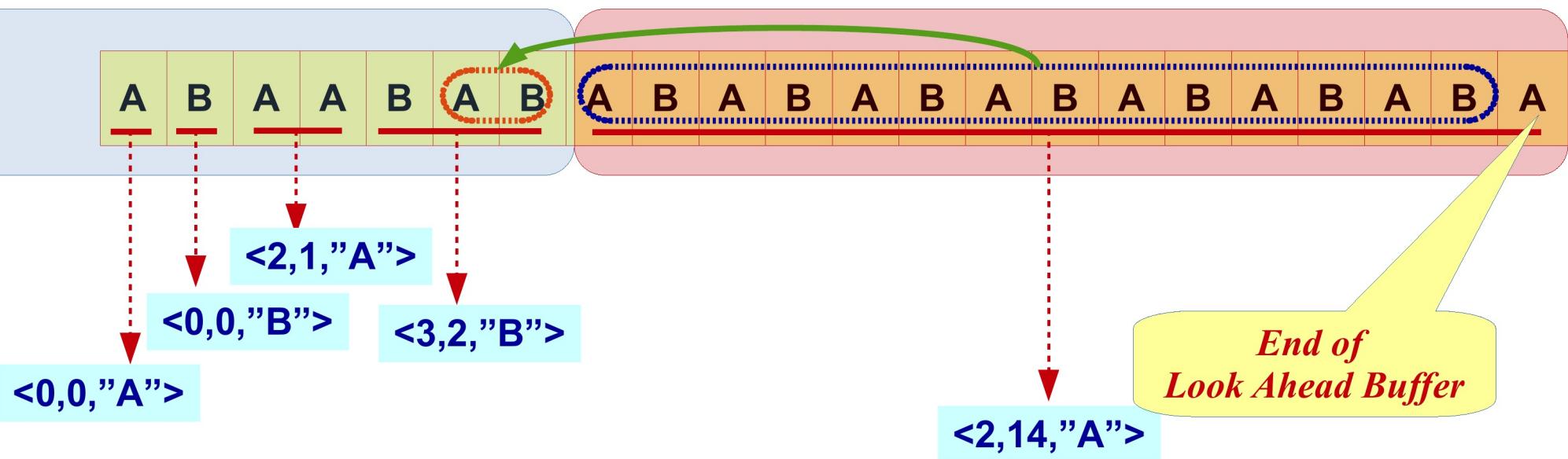
# LZ 77 (Compression) (Handling Repetitive Sequence)



*There are **Four** Consecutive Symbols “BA” in Look Ahead Buffer  
“BA” exists in search buffer Adjacent to Look Ahead Buffer  
Go Back **Two Steps**, Pick **Four Symbols**  
**Position=2, Length =4, next Symbol=NULL***

# LZ 77 (Compression) (Handling Repetitive Sequence)

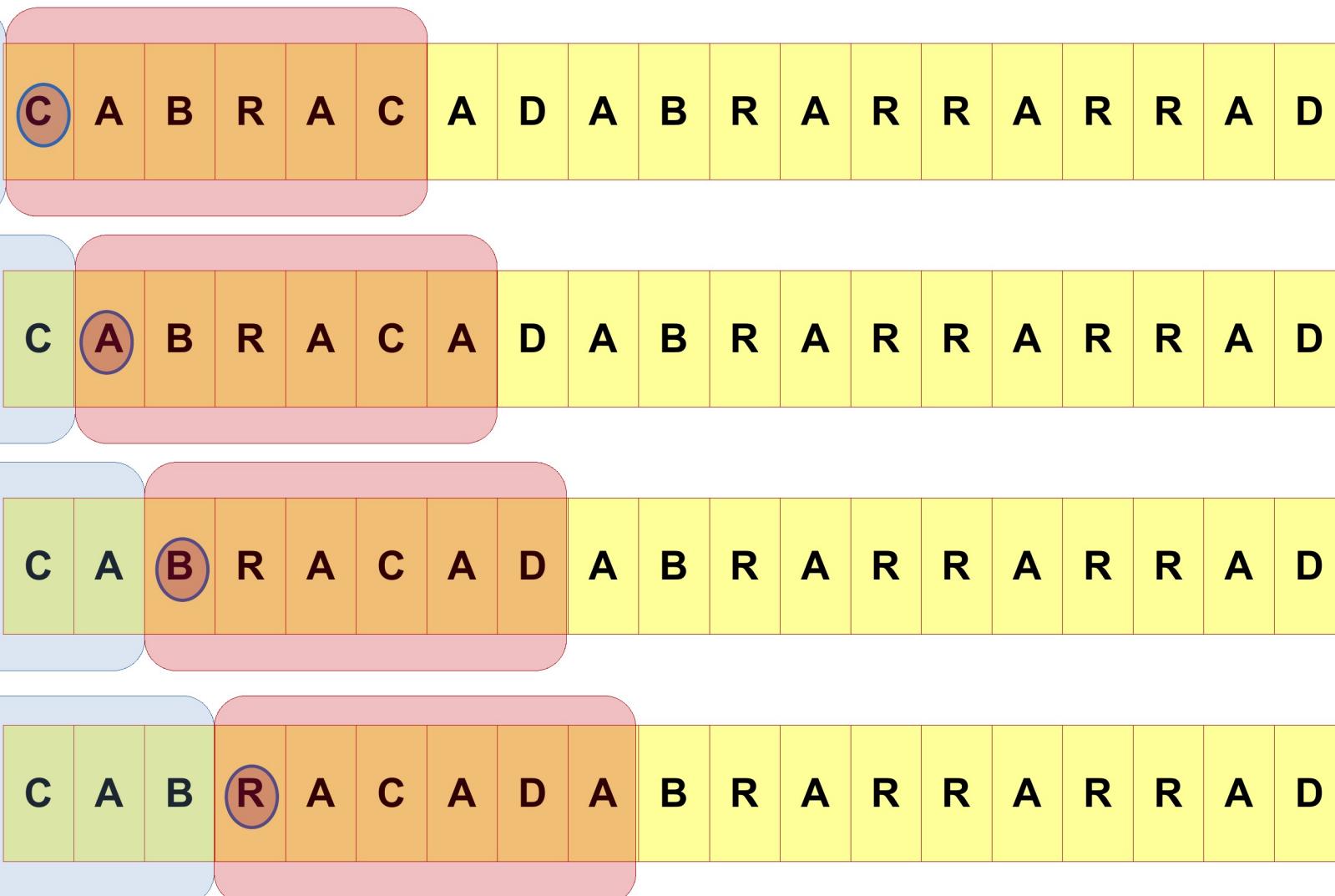
*What if we use BIGGER look Ahead Buffer ?*



*There are 14 Consecutive Symbols "AB" in Look Ahead Buffer  
"AB" exists in search buffer Adjacent to Look Ahead Buffer  
Go Back Two Steps, Pick Ten Symbols  
Position=2, Length =14, next Symbol="A"*

# LZ 77 (Compression)

(Handling Repetitive Overlapped Sequence)



<0,0,"C">

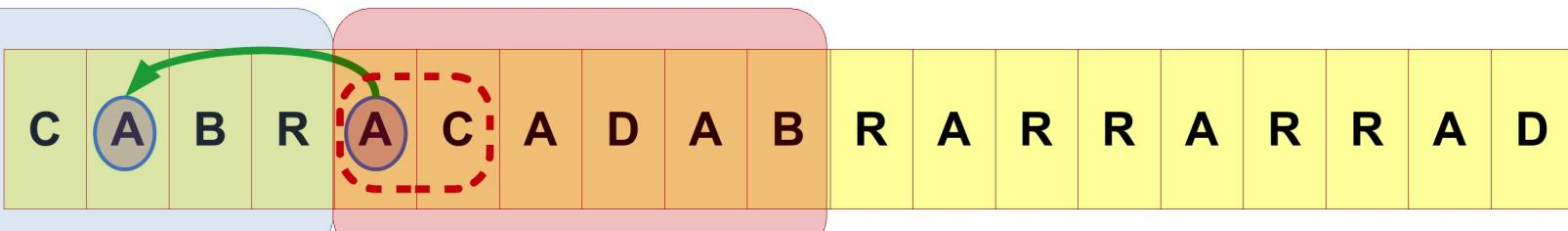
<0,0,"A">

<0,0,"B">

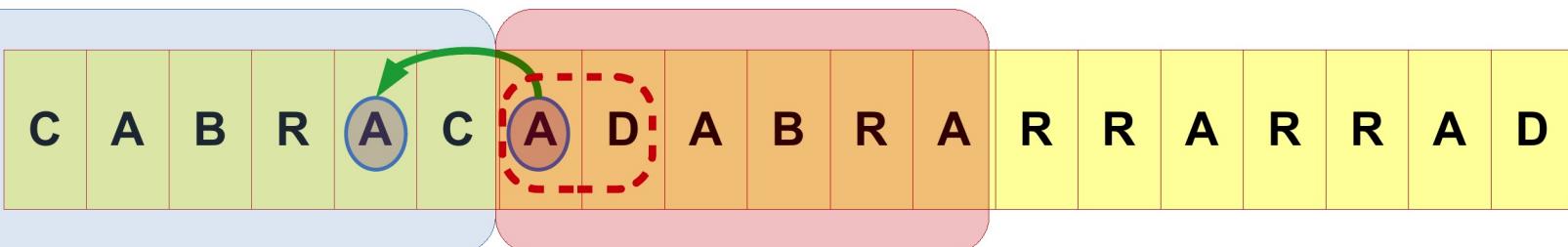
<0,0,"R">

# LZ 77 (Compression)

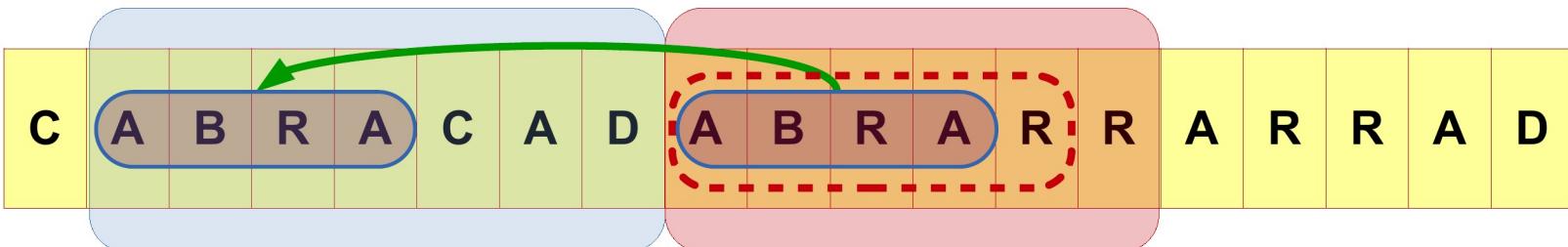
(Handling Repetitive Overlapped Sequence)



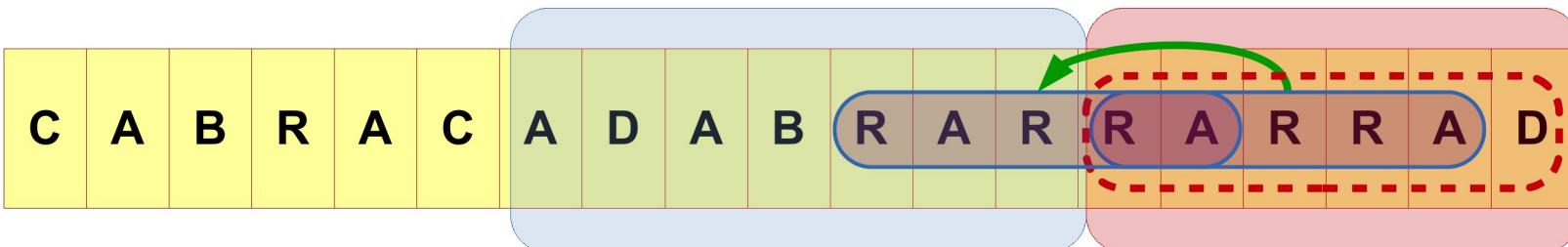
<3,1,"C">



<2,1,"D">



<7,4,"R">



<3,5,"D">



R A P

R A R R R A D

<3, 5, '0'>



# Compression Ratio

**<0,0,"C">**

**<0,0,"A">**

**<0,0,"B">**

**<0,0,"R">**

**<3,1,"C">**

**<2,1,"D">**

**<7,4,"R">**

**<3,5,"D">**

**Original Size** = Number of Symbols \* Bits used to Store one Symbol  
= **19 Symbols \* 8 Bits / Symbol = 152 bits**  
(Store “Symbol” ASCII Code in 8 Bits)

**Max “Position” Value = 7**

**Max “Length” Value=5**

**Max Symbols = 256 Symbol**

**Tag size = 3 + 3 + 8 =14 Bits**

Store “Position” Value in **3 Bits**

Store “Length” Value in **3 Bits**

Store “Symbol” ASCII Code in 8 Bits

**Number of Tags = 8 Tags**

**Compressed Size=8\*14=112 bits**

# Advantages and Disadvantage of LZ77

## Advantages of LZ77

- Probabilities of symbols are not required to be known a priori. (suitable for Real time Compression).
- That is, the longer the size of the sliding window, the better the performance of data compression
- No coding table Required for Decompression.

## Disadvantage of LZ77

- A straightforward implementation would require up to  $[\text{Look Ahead Buffer Size}] * [\text{Search Window Size}]$  Symbols comparisons per Tag produced.
- Complexity of comparison is very large