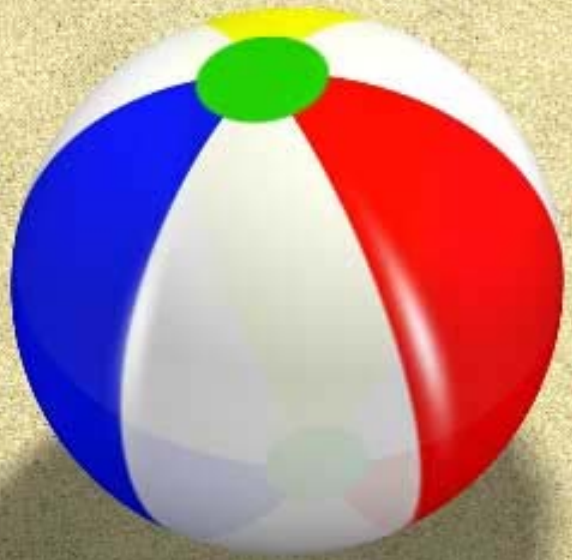
Simpson's Hidden Talents



Problem

- Given two strings S_1 and S_2 find the longest prefix of S_1 that is a suffix of S_2

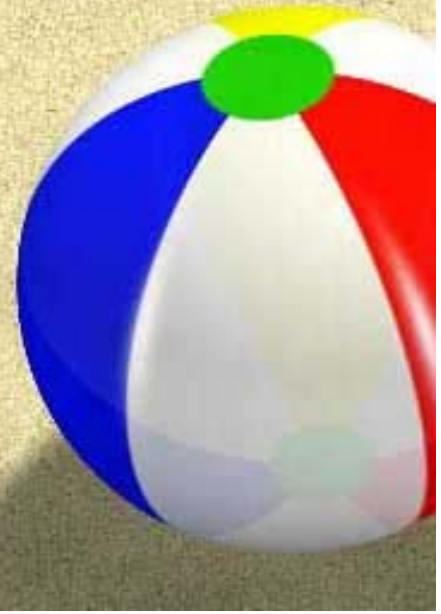
Problem

- Given two strings S_1 and S_2 find the longest prefix of S_1 that is a suffix of S_2

- Example:

S_1 : ababcababdaba

S_2 : babacbadababc



Problem

- Given two strings S_1 and S_2 find the longest prefix of S_1 that is a suffix of S_2

- Example:

S_1 : ababcababdaba

S_2 : babacbadababc

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : b a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

×

S_2 : b a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

×

S_2 : b a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

×

S_2 : c b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

×

S_2 : b a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

×

S_2 : d a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

First Idea

S_1 : a b a b c a b a b d a b a

S_2 : a b a b c

Answer

What if ...

S_1 : a a a a a a a a a a a a a a

S_2 : a a a a a a a a a a a a a z

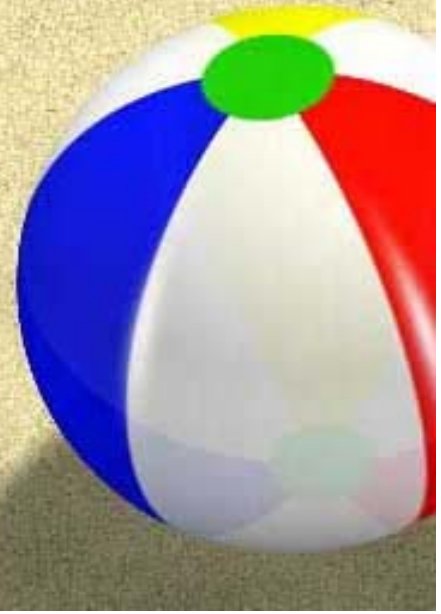
What if ...

S_1 : a a a a a a a a a a a a a a

S_2 : a a a a a a a a a a a a a z

How many comparisons?

$$n + (n-1) + \ldots + 2 + 1 = \Theta(n^2)$$

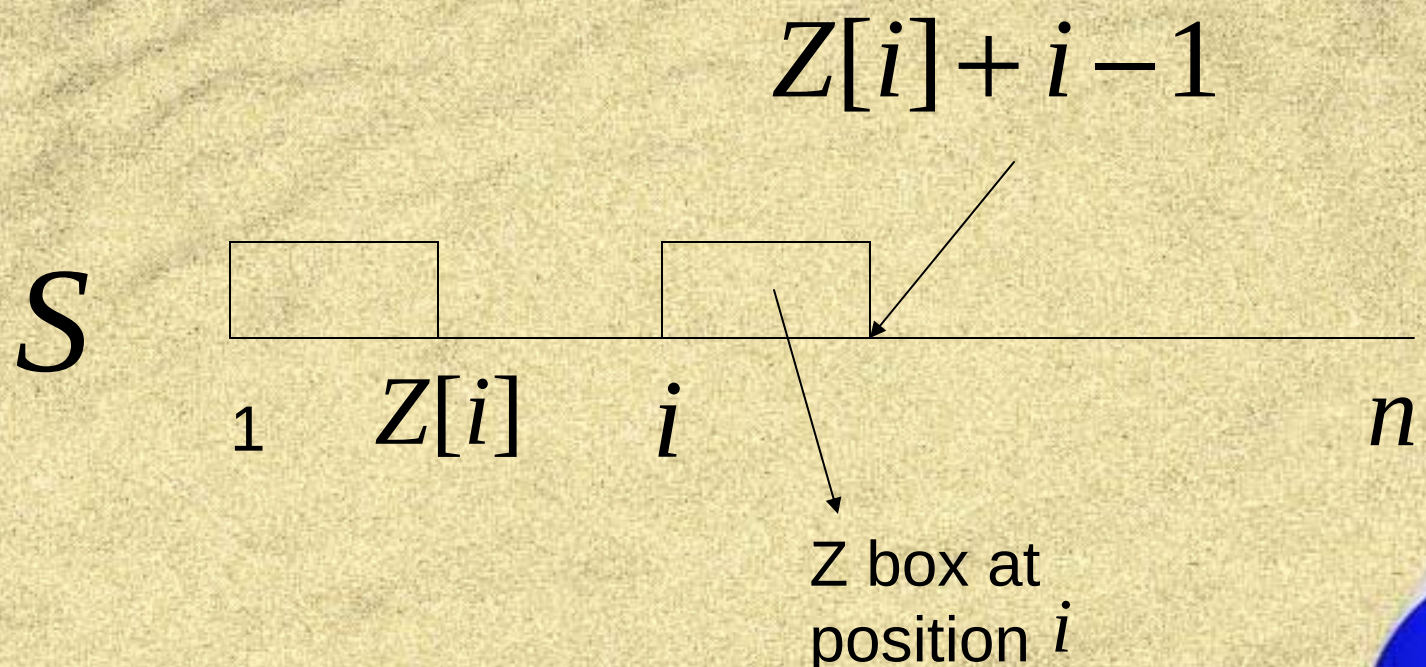


Recall the Z algorithm*

- Simplest linear string matching algorithm
- Given a string $S[1..n]$, denote by $Z[i]$ the length of the longest string starting at position i of S, that is a prefix of S
  - Note: just one string
- Compute $Z[i]$ iteratively

*Algorithms on strings, trees and sequences, D.Gusfield

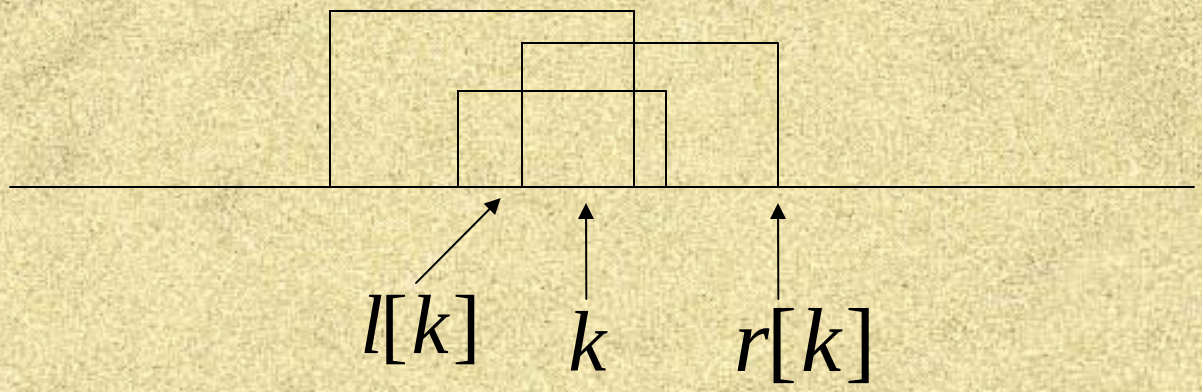
Definitions



Definitions

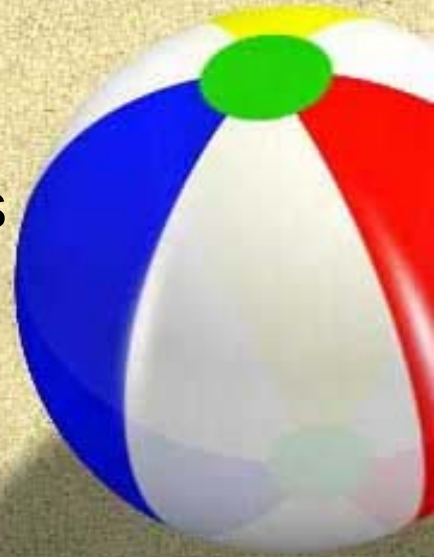
Fix a position k

S



$r[k]$ Rightmost end of a Z box that starts at a position $1 < j \leq k$

$l[k]$ The left end of the Z box defined by $r[k]$



The Z Algorithm

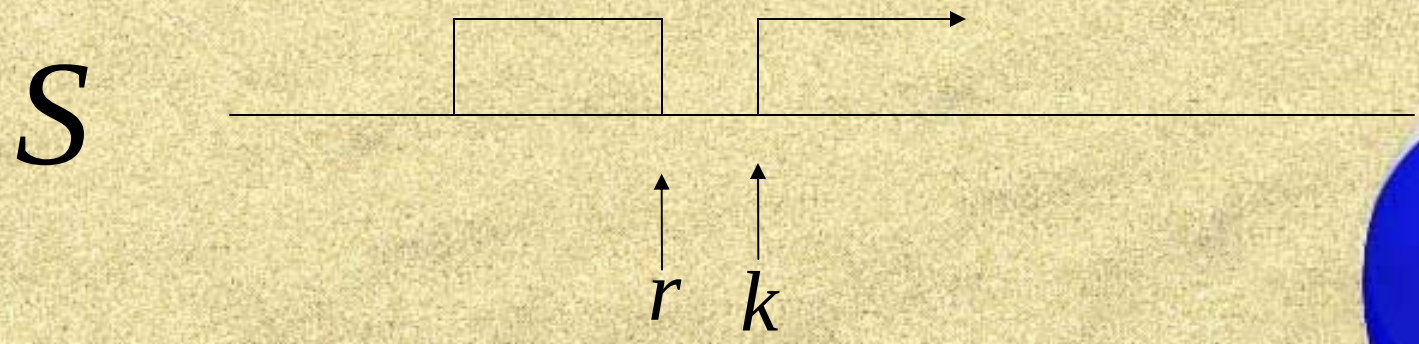
- Compute iteratively $Z[i], l[i], r[i]$
- Only last value of $l[i], r[i]$ will be needed. Refer to them as l, r
- Three cases to consider

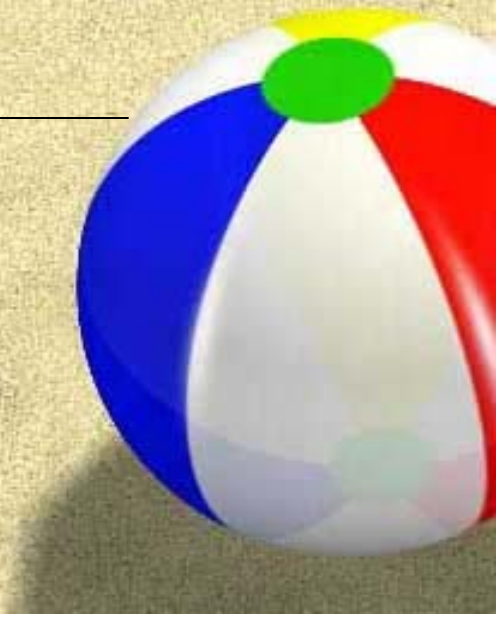
The Z Algorithm

Iteration k

Case 1: $k > r$

Compute $Z[k]$ by explicitely comparing the characters at position k and on with the characters at the beginning of the string

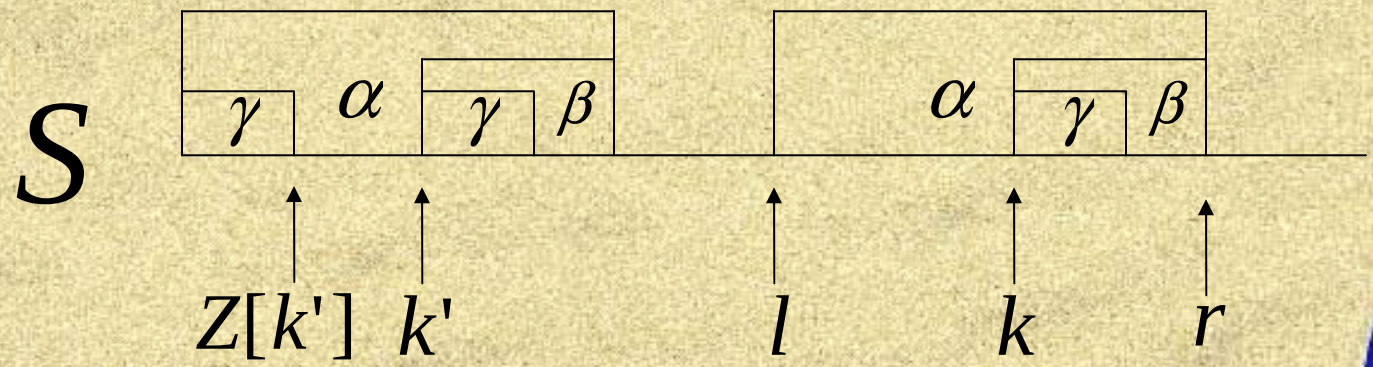




The Z Algorithm

Iteration $k, k \leq r$

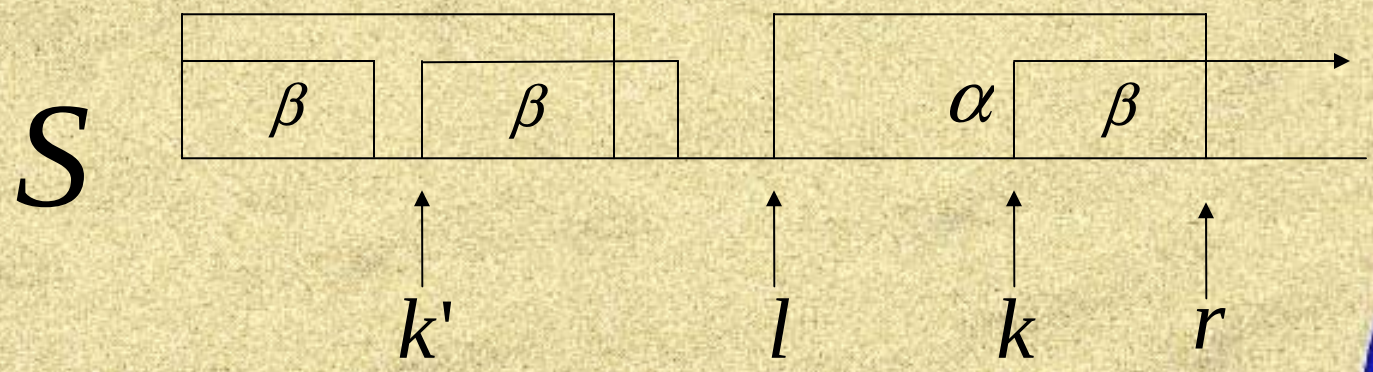
Case 2a: $Z[k'] < |\beta|$



The Z Algorithm

Iteration $k, k \leq r$

Case 2b: $Z[k'] \geq |\beta|$



The Z algorithm

- Why does it run in linear time?

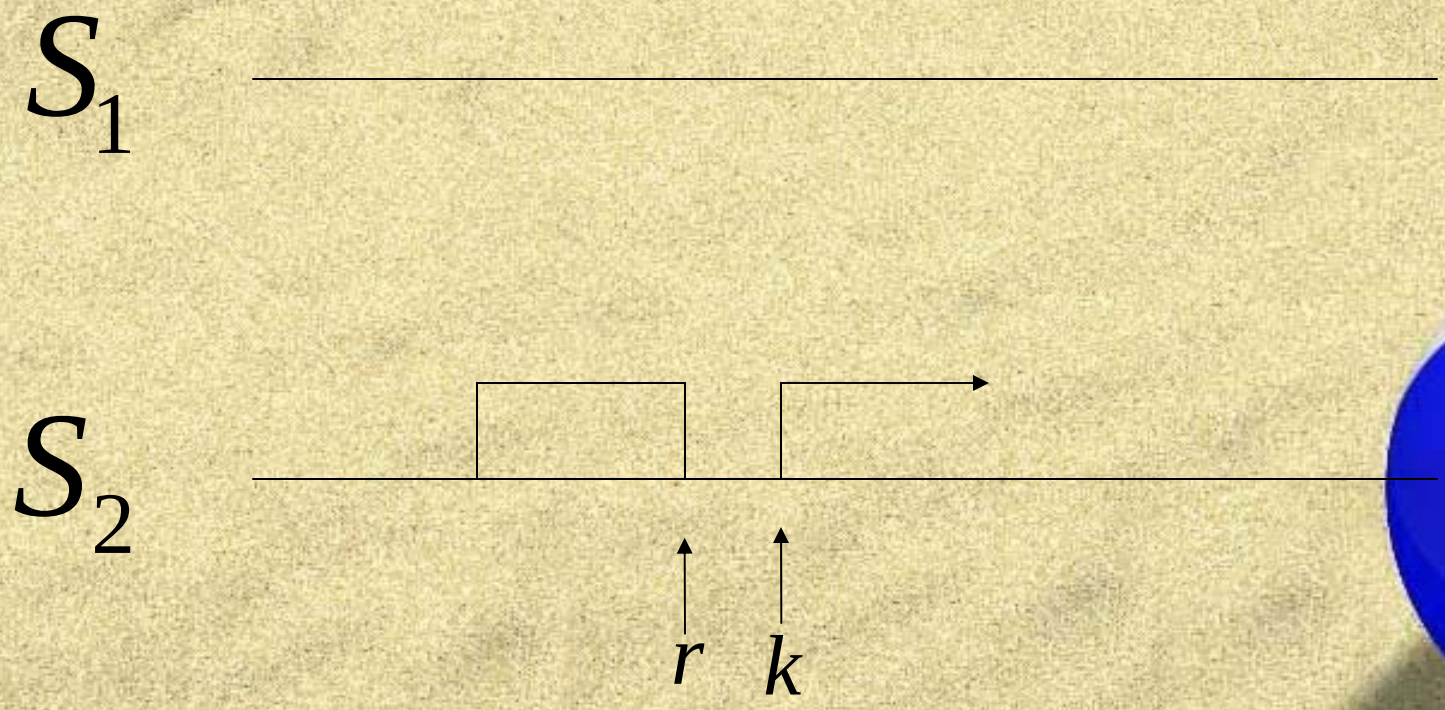
The Contest Problem

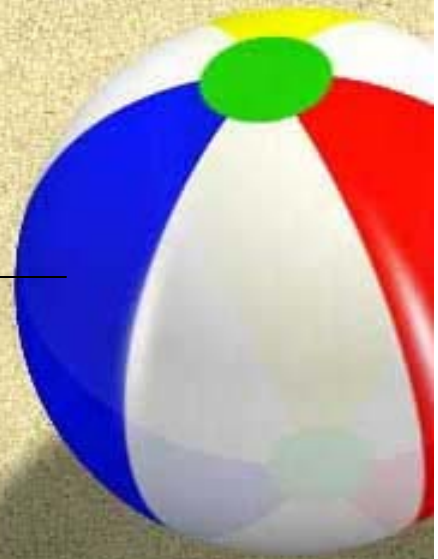
- What if we have two strings?
- The solution is a simple extension of the Z algorithm
- Call $Y[i]$ the longest string which starts at position i of S_2 and which is also a prefix of S_1

The Contest Problem

- Definition of a Y box, l and r are analogous, this time with respect to S_2

Case 1

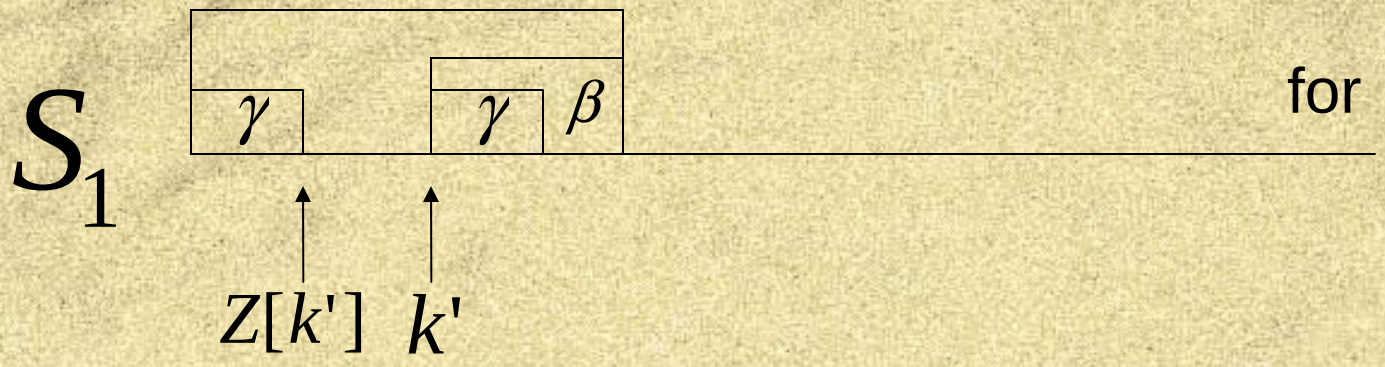


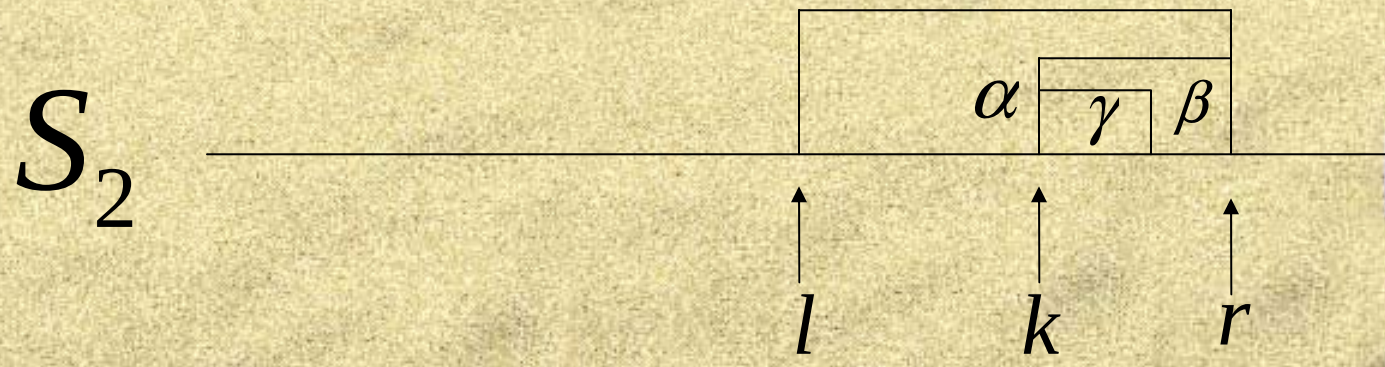


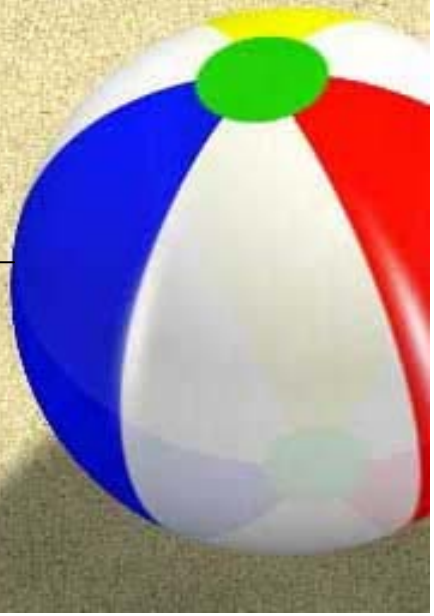
Case 2a

$Z[k'] < |\beta|$

Need Z values for S_1







Case 2b

$Z[k'] \geq |\beta|$