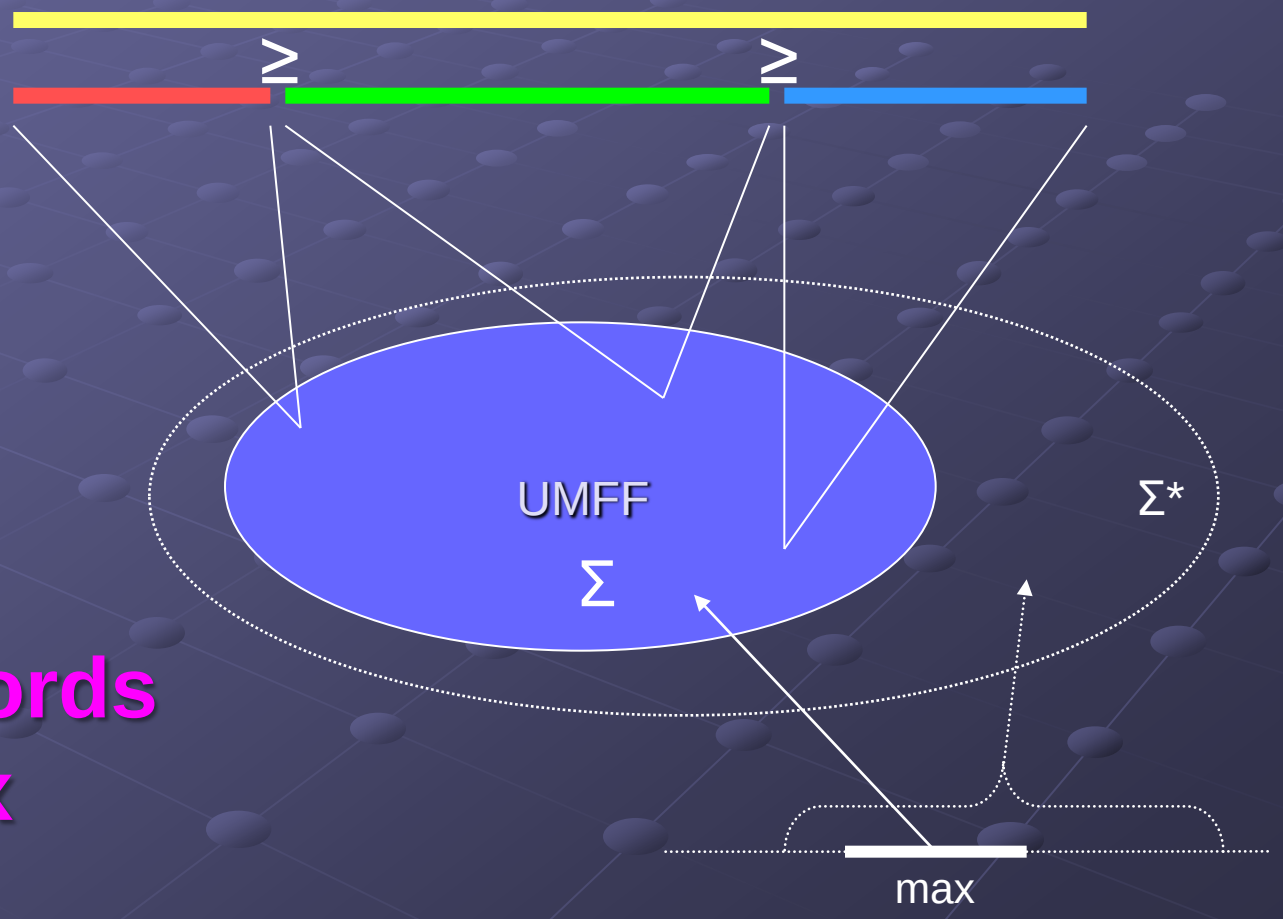
String Comparison & Lyndon-Like Factorization Using V-Order in Linear Time

David & Jackie Daykin and Bill Smyth

Institute of mathematics & its applications

UMFF

- Unique Maximal Factorization Family



Lyndon words

L = xy < yx

Types of UMFFs

- Definition. An UMFF W is type Flight Deck iff u[1...n] in W implies that $u[1] \leq_W u[i]$

- Definition. An UMFF is type Acrobat iff it contains elements uv_1, w and uv_2, nonempty u not a prefix of w, such that $uv_1 <_W w <_W uv_2$

Hybrid Lyndons

- Definition. A word $u = u_1 u_2 \ldots u_n$

$$= u_1 U_1 u_1 U_2 \ldots u_1 U_m$$

where u_1 is a letter and $U_1, U_2, \ldots, U_m$ are strings (possibly empty) is a Hybrid Lyndon under lex extension $\leq_T$ iff $U_1 U_2 \ldots U_m$ is Lyndon according to a total order T

Example: Lyndon = lex extension + lex order!

V-order ≺

- Strehl, Winkelmann 1981; Dahn, D. Daykin 1996
- Delete ‘V’ elements until equal; apply co-lex order
- If w′ is any proper subsequence of w, w′ ≺ w

x = 9199566 y = 9199665

x ≻ y

Function: delete V O(n)

V-words

- Definition. A word v is a *V-word* if it is the minimum in V-order over all rotations of v

- Theorem. A string v is a V-word iff it is a Hybrid Lyndon using V-order

- Theorem. The set of V-words forms an UMFF
  - *the V-UMFF is in V-order* ‹ *and* ›
  - *the V-UMFF is type Flight Deck*

→ V-words ≠ Lyndons

V-factorization

- Lemma. Deciding V-order is linear

- Algorithm = Lyndon + V-ordering $O(n^2)$
- Optimize via concatenation of factors $O(n)$

1. 513341229163 linear sweep
2. 99999??99??? 99921999 12

frequency of max letters

apply V-ordering

Linear V-comparison

- while $|u| \neq |v|$

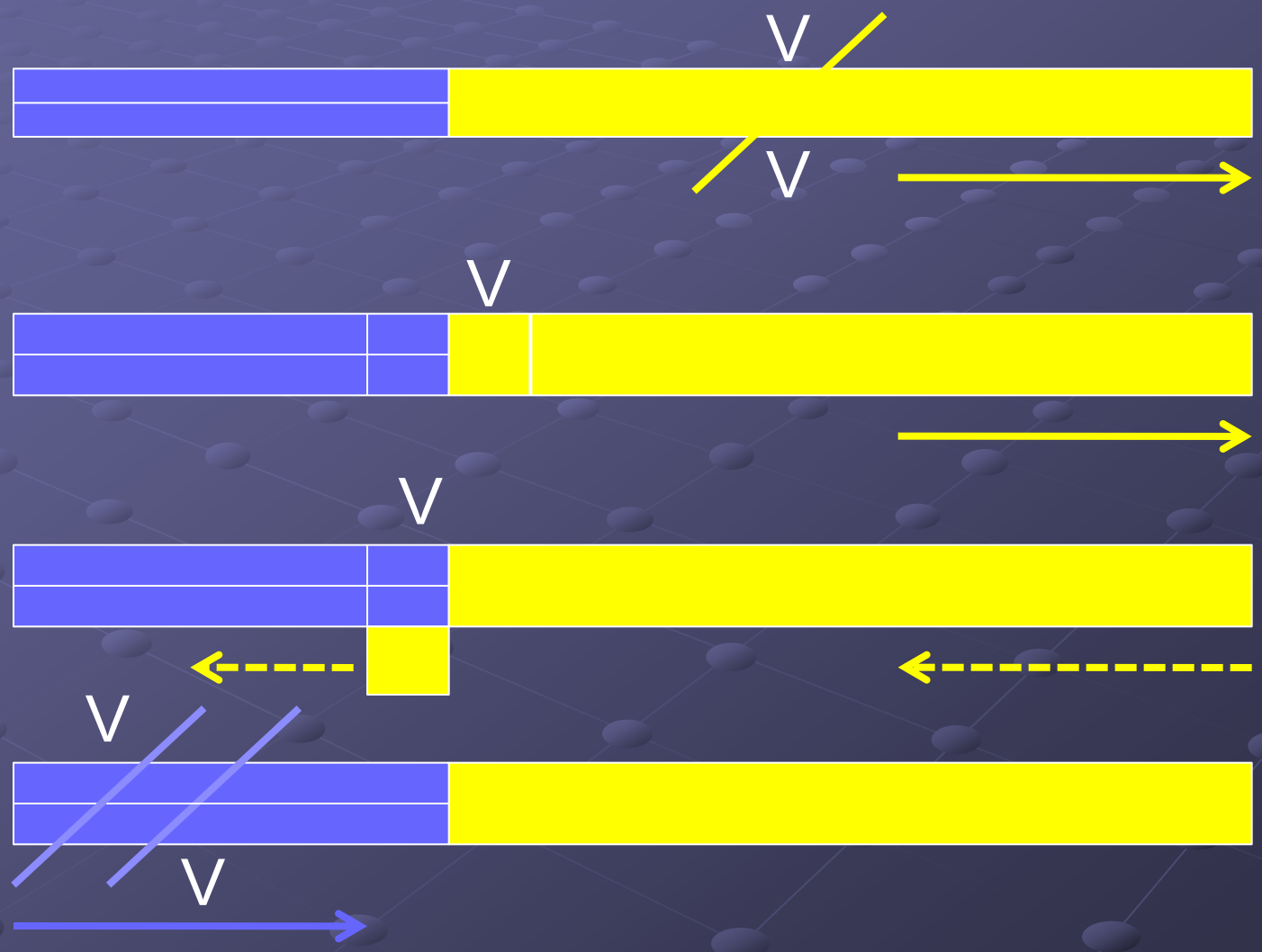
  delete V letter of longer string

- while $u \neq v$

  compute Longest Matching Suffix and delete V letter

- when LMS = reduced strings, decide $\prec$ order using last V deletions

Longest Matching Suffix



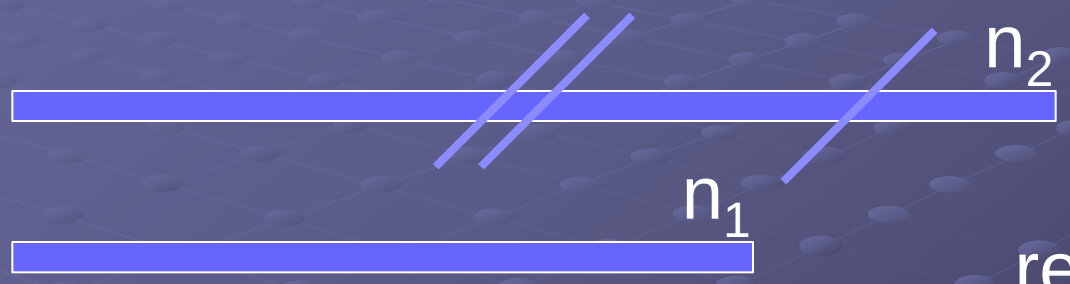
Data Structure Θ(n)

- Represent strings with a doubly-linked list

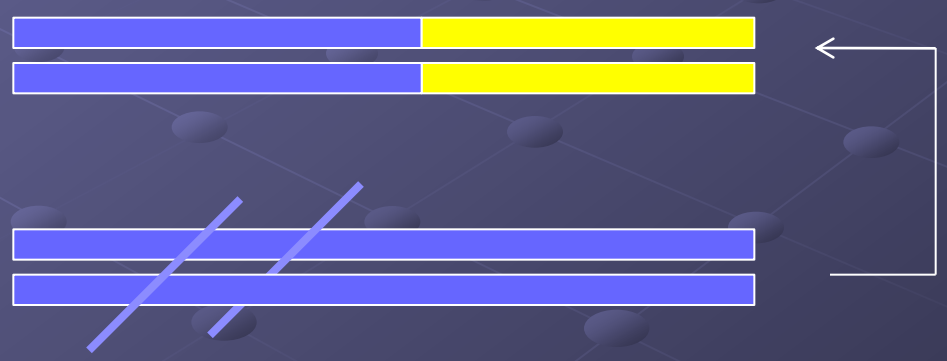
- After V deletion update pointers

- Operations – read / write O(1)

Complexity $\Theta(n_1+n_2)$



reduce $\Theta(n_2)$



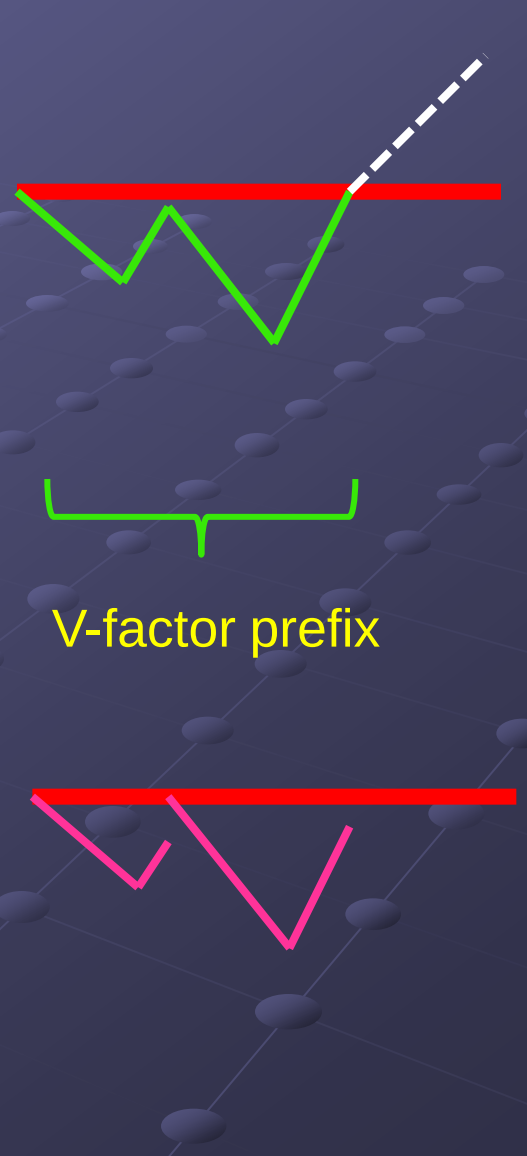
LMS(?)

$\Theta(n_1)$

delete V

Duval's algorithm for V-words $O(n^2)$

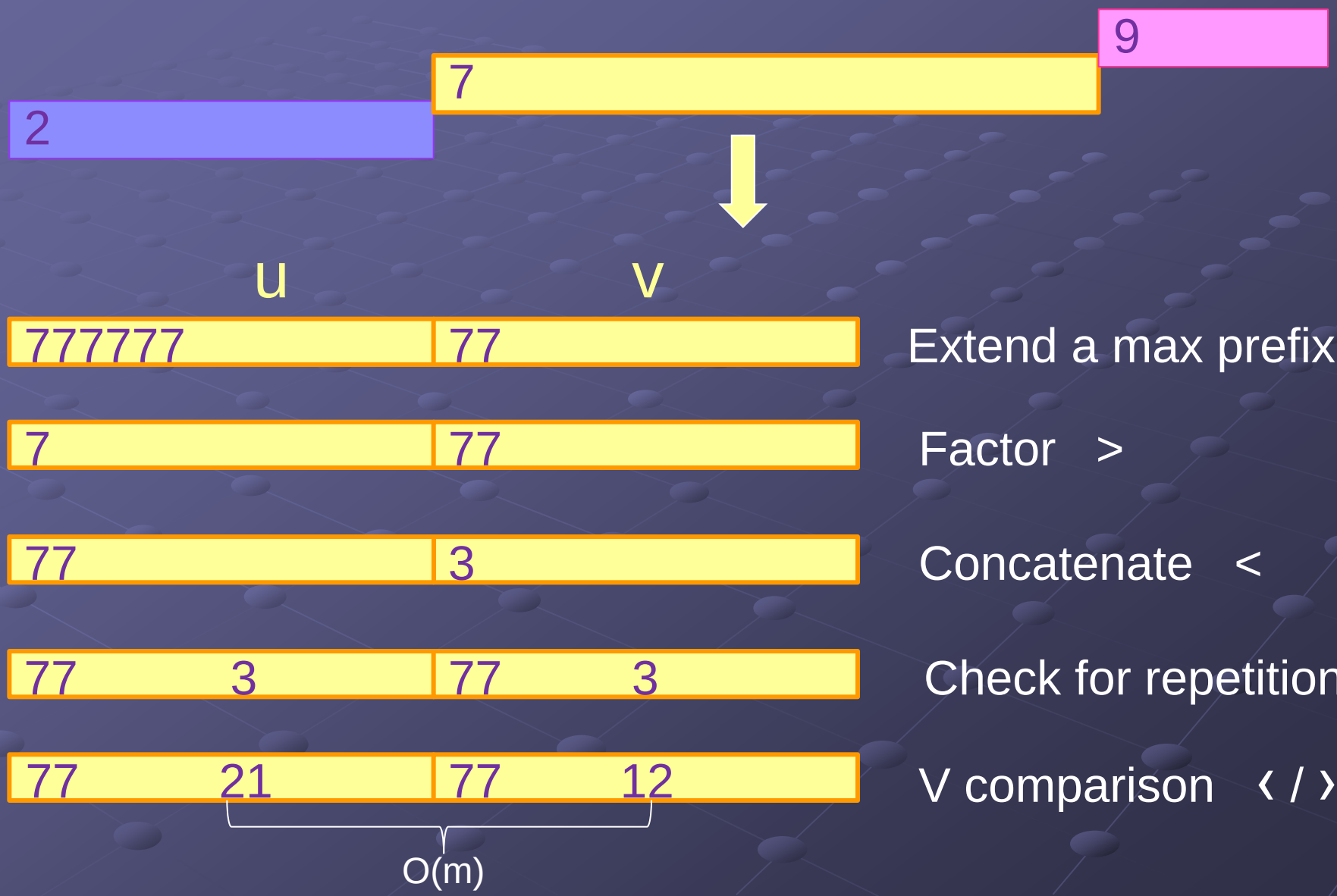
```
input x[1..n]
j ← 1; l ← 2
repeat
   while x[j] ≥ x[l] and l ≤ n do l ← l + 1;
   LynV(x[j..l -1]); j ← l; l ← l + 1
until j > n
LynV(x[1..m])
h ← 0
while h < m do
i1 ← h + 1; i2 ← end(i1); j1 ← i2 + 1; j2 ← end(j1)
while x[j1..j2] ≥ x[i1..i2] do
  if x[j1..j2] > x[i1..i2] then i1 ← h + 1; i2 ← end(i1)
     else i1 ← i2+1; i2 ← end(i1)
  j1 ← j2 + 1; j2 ← end(j1)
repeat h ← h + (j1 - i1); output h until h ≥ i1
```



V-word factorization VF(x)

```
h ← 0
while h < n do
RESET
while x[j] ≤ g do
if x[i] = x[j] = g then
if j-i = k1 then k1 ← k1+1;
rep ← rep+1; l1 ← l1+1
else
if i-h ≤ k1 then k2 ← k2+1
l2 ← l2+1; i ← i+1
elsif x[j] = g then
output (h, rep, l1); RESET; j ← j-1
elsif x[i] = g then
if j = k1+1 then l1 ← k1+1
else l1 ← (rep x l1)+1
rep ← 1; i ← h+1
if k2 > 0 then l1 ← l1+l2; k2, l2 ← 0
elsif x[i] = x[j] then
l2 ← l2+1
if l2 = l1 then rep ← rep+1; k2, l2 ← 0; i ← h+1
else i ← i+1
else
j' ← j; while x[j+1] < g do j ← j+1
if not prec(x[h+1..h+l1], l1, x[j'-l2..j], j-j'+l2+1) then
output (h, rep, l1)
l1 ← j-h; rep ←1; k2, l2 ← 0; i ← h+1
j ← j+1
output (h, rep, l1) ; output (h, 1, l2)
```

VF(x) concatenation



Complexity O(n)

- Worst-case: V-compare and concatenate

$$x = \underbrace{\underbrace{\underbrace{\underbrace{9192}\ 9193}\ 91929194}\ 91929195}$$

Assume $|x| = 2^m$

Comparisons $2 + \ldots + 2^{m-1} + 2^m = 2(|x| - 1)$

Hybrid Lyndon theory

- Lyndon words shuffle (Cummings & Mayes 1992)

  V-shuffle *ddbadcdcdbdcb*

- V-prefix < V-word < suffix

- Concatenation order

  xy < xyyzxyz < yz Lyndon

  xy < xyzxyyz < yz co-Lyndon

  V-words

Applications ?

- Optimization problems:

Lyndon factorization: 443427172998

V-words: 443427172998

linear

- Musicology:

African rhythms (Chemillier & Truchet 2003)

Rhythmic V-words	Ethnic group
332	*Zande*
32322	*Aka, Gbaya, Nzakara*
3223222	*Gbaya, Ngbaka*
322232222	
32222322222	*Aka*

Acknowledgements

Bill Smyth

David Daykin

Dedication

This talk is dedicated to the loving memory of:
David Daykin – founder of UMFF frolics!