Language Features of version 18.0 in Depth

Adám Brudzewsky

New

`⎕C`	Case convert
`f⍤⍥g`	Over
`f⍤⍤g`	Atop
`≠Y`	Unique mask
`A⍨`	Constant
`⎕DT`	Date-time
`1200⌶`	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

New

`⎕C`	Case convert
`f⍤⍥g`	Over
`f⍤⍤g`	Atop
`≠Y`	Unique mask
`A⍨`	Constant
`⎕DT`	Date-time
`1200⌶`	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
```

dyalog.tv/webinar

New

⎕C	Case convert
f⍤⍥g	Over
f⍤⍤g	Atop
≠Y	Unique mask
A⍨	Constant
⎕DT	Date-time
1200⌶	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Turn boxing on

```
]box on
```

Partitioned Enclose with Empties

everything is possible

X⊂Y

Evolution

```
      1 0 0 1 1 0 0 0 0 0 0 ⊂ 'KenEIverson'
```

Evolution

```
      1 0 0 1 1 0 0 0 0 0 0 ⊂ 'KenEIverson'
```


```
      1 0 0 2 2 0 0 0 0 0 0 ⊂ 'KenEIverson'
```

Use case: partitioning into bins

```
cutoffs ← 0 20 40 60 80 100
```

Use case: partitioning into bins

```
cutoffs ← 0 20 40 60 80 100

values ← 3 14 15 35 65 89 92 793
```

Use case: partitioning into bins

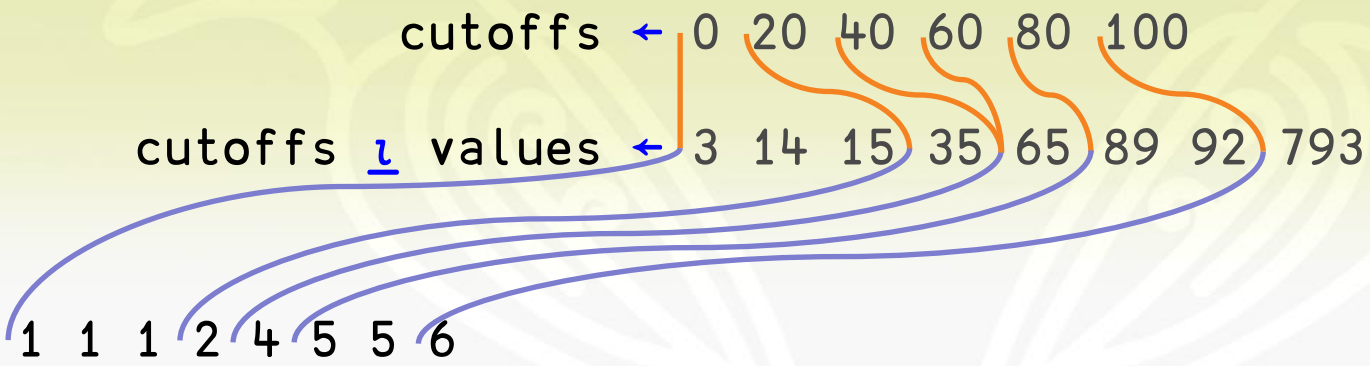
```
      cutoffs ← 0 20 40 60 80 100

cutoffs ⍸ values ← 3 14 15 35 65 89 92 793

1 1 1 2 4 5 5 6
```

Use case: partitioning into bins

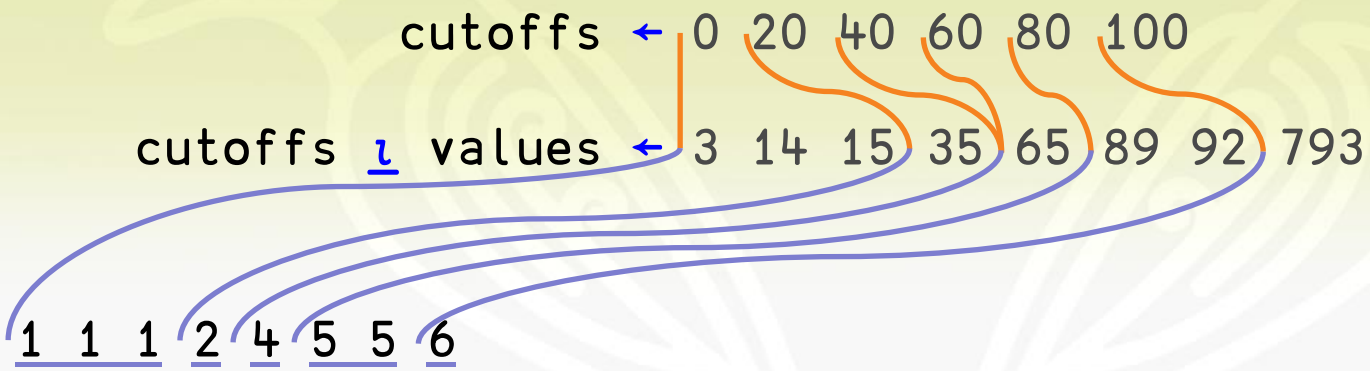
```
      cutoffs ← 0 20 40 60 80 100

cutoffs ⍸ values ← 3 14 15 35 65 89 92 793
```

```
1 1 1 2 4 5 5 6
```

Use case: partitioning into bins



Use case: partitioning into bins



Use case: partitioning into bins

```
      cutoffs ← 0 20 40 60 80 100

cutoffs ⍸ values ← 3 14 15 35 65 89 92 793
1 1 1 2 4 5 5 6
```

Use case: partitioning into bins

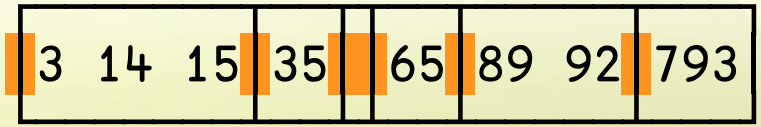
```
      cutoffs ← 0 20 40 60 80 100

cutoffs ⍸ values ← 3 14 15 35 65 89 92 793
1 1 1 2 4 5 5 6
      ¯2-/ cutoffs ⍸ values
0 0 1 2 1 0 1
```

Use case: partitioning into bins

```
      cutoffs ← 0 20 40 60 80 100

      cutoffs ⍸ values ← 3 14 15 35 65 89 92 793
1 1 1 2 4 5 5 6
      ¯2-/ cutoffs ⍸ values
0 0 1 2 1 0 1
      1, ¯2-/ cutoffs ⍸ values
1 0 0 1 2 1 0 1
```

Use case: partitioning into bins

```
      cutoffs ← 0 20 40 60 80 100

      cutoffs ⍸ values ← 3 14 15 35 65 89 92 793
1 1 1 2 4 5 5 6
      ⍳2-/ cutoffs ⍸ values
0 0 1 2 1 0 1
      1, ⍳2-/ cutoffs ⍸ values
1 0 0 1 2 1 0 1
      values ⊂⍨ 1, ⍳2-/ cutoffs ⍸ values
```



Use case: partitioning into bins

```
      cutoffs ← 0 20 40 60 80 100

      cutoffs ⍸ values ← 3 14 15 35 65 89 92 793
1 1 1 2 4 5 5 6
      ⍳2-/ cutoffs ⍸ values
0 0 1 2 1 0 1
      1, ⍳2-/ cutoffs ⍸ values
1 0 0 1 2 1 0 1
      values ⊂⍤1, ⍳2-/ cutoffs ⍸ values
      cutoffs ( ⊢ ⊂⍤1, ⍳2-/ ⍸ ) values
```

Use case: partitioning into bins

```
      BinK ← 1↓¨,⊂⍤⊢⌸⍨⍳⍤≢⍤⊣,⍸ ⋄ BinI←⊢⊂⍨1,¯2-/⍸
      c←10×⍳10 ⋄ v←10+10÷⍨{⍵[⍋⍵]}?⍴⍨1000

      ]runtime -c "c BinK v" "c BinI v"
c BinK v → 1.7E¯5 |    0% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
c BinI v → 4.2E¯6 | -76% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
```

Use case: partitioning at given indices

```
data ← 'KenEIverson'
inds ←  1  4 5
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ←  1  4 5
      ⎕← mask ← ⍸⍣¯1 ⊢ inds
1 0 0 1 1
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ←  1  4 5
      ⎕← mask ← ⍸⍣¯1 ⊢ inds
1 0 0 1 1
      ↑ data mask
K e n E I v e r s o n
1 0 0 1 1 0 0 0 0 0 0
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ← 1 4 5
      ⎕← mask ← ⍸⍣¯1 ⊢ inds
1 0 0 1 1
      ↑ data mask
K e n E I v e r s o n
1 0 0 1 1 0 0 0 0 0 0
      mask ⊂ data
┌───┬─┬───────┐
│Ken│E│Iverson│
└───┴─┴───────┘
```

Use case: partitioning at given indices

```
data ← 'KenEIverson'
inds ←  1 4 4 5 5
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ←  1 4 4 5 5
      ⎕← mask ← ⍸⍣¯1 ⊢ inds
1 0 0 2 2
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ←  1 4 4 5 5
      ⎕← mask ← ⍸⍣¯1 ⊢ inds
1 0 0 2 2
      mask ⊂ data
┌───┬┬─┬┬───────┐
│Ken││E││Iverson│
└───┴┴─┴┴───────┘
```

Use case: partitioning at given indices

```
data ← 'KenEIverson'
inds ←  1 1 1 4 4 5 5,1+≠data
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ←  1 1 1 4 4 5 5,1+≠data
      ⎕← mask ← ⍳⍤¯1 ⊢ inds
3 0 0 2 2 0 0 0 0 0 0 1
```

Use case: partitioning at given indices

```
      data ← 'KenEIverson'
      inds ←  1 1 1 4 4 5 5,1+≢data
      ⎕← mask ← ⍸⍣¯1 ⊢ inds
3 0 0 2 2 0 0 0 0 0 0 1
      mask ⊂ data
┌┬┬───┬┬─┬┬───────┬┐
││││Ken││E││Iverson││
└┴┴───┴┴─┴┴───────┴┘
```

Extensions to Mix with Axis

Hello, APL2ers!

```
↑[k]Y
```

Set boxing to max

```
]box -style=max
```

Set boxing to max

```
]box -style=max
```

Set boxing to max

```
]box -style=max
```

Max boxing

3 ⍴ □A

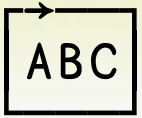
Max boxing

```
      3⍴⎕A
┌→──┐
│ABC│
└───┘
```

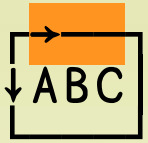
```
1     3⍴⎕A
┌→──┐
↓ABC│
└───┘
```

Max boxing

3⍴⊂A



1 3⍴⊂A



Max boxing

```
      3⍴⎕A
┌→──┐
│ABC│
└───┘
      2  1  3⍴⎕A
┌┌→──┐
↓↓ABC│
││   │
││DEF│
└└───┘
      1  3⍴⎕A
┌→──┐
↓ABC│
└───┘
```

Max boxing

```
      1   3⍴⎕A
┌→──┐
↓ABC│
└───┘
```

```
      3⍴⎕A
┌→──┐
│ABC│
└───┘
```

```
      2 1   3⍴⎕A
 ┌→──┐
↓↓ABC│
 └───┘
 DEF
```

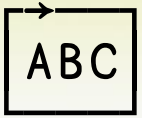
Max boxing

```
      3⍴⎕A
┌→──┐
│ABC│
└───┘
      2 1 3⍴⎕A
┌┌→──┐
↓↓ABC│
││   │
││DEF│
└└───┘
```

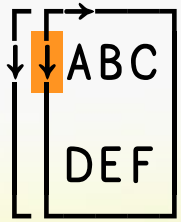
```
      1 3⍴⎕A
┌→──┐
↓ABC│
└───┘
```

Max boxing

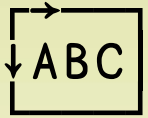
```
      3⍴⊂'ABC'
```




```
      2 1 3⍴⊂'ABC'
```




```
      1 3⍴⊂'ABC'
```



Max boxing

```
      3ρ⎕A
┌→──┐
│ABC│
└───┘
```

```
      1 3ρ⎕A
┌→──┐
↓ABC│
└───┘
```

```
      2 1 3ρ⎕A
┌┌→──┐
↓↓ABC│
││   │
││DEF│
└└───┘
```

Max boxing

```
      3ρ⎕A
┌→──┐
│ABC│
└───┘
      2 1 3ρ⎕A
┌┌→──┐
↓↓ABC│
││   │
││DEF│
└└───┘
```

```
      1 3ρ⎕A
┌→──┐
↓ABC│
└───┘
```

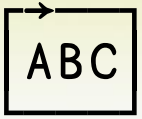
Max boxing

```
      3⍴⎕A
┌→──┐
│ABC│
└───┘
      2 1 3⍴⎕A
┌┌→──┐
↓↓ABC│
││   │
││DEF│
└└───┘
```

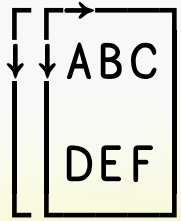
```
      1 3⍴⎕A
┌→──┐
↓ABC│
└───┘
      2 2 1 3⍴⎕A
┌┌┌→──┐
↓↓↓ABC│
│││   │
│││DEF│
│││   │
│││   │
│││GHI│
│││   │
│││JKL│
└└└───┘
```

Max boxing

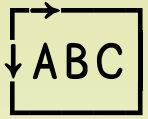
3ρ⊂A



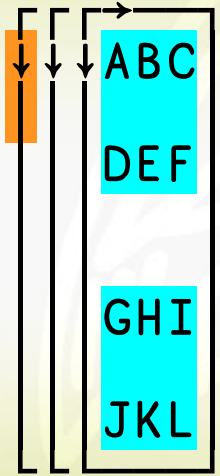
2 1 3ρ⊂A



1 3ρ⊂A

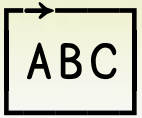


2 2 1 3ρ⊂A

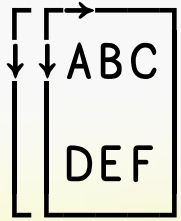


Max boxing

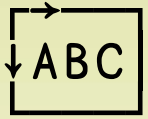
3⍴⊂A



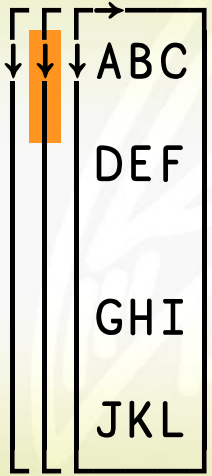
2 1 3⍴⊂A



1 3⍴⊂A

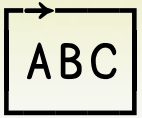


2 2 1 3⍴⊂A

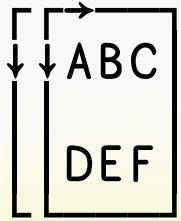


Max boxing

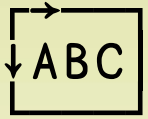
3⍴⊂A



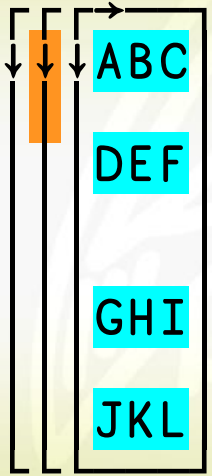
2 1 3⍴⊂A



1 3⍴⊂A



2 2 1 3⍴⊂A



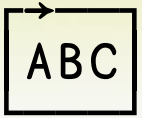
Max boxing

```
      3ρ⎕A
┌→──┐
│ABC│
└───┘
      2 1 3ρ⎕A
┌┌→──┐
↓↓ABC│
││   │
││DEF│
└└───┘
```

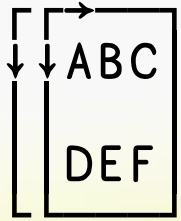
```
      1 3ρ⎕A
┌→──┐
↓ABC│
└───┘
      2 2 1 3ρ⎕A
┌┌┌→──┐
↓↓↓ABC│
│││   │
│││DEF│
│││   │
│││   │
│││GHI│
│││   │
│││JKL│
└└└───┘
```

Max boxing

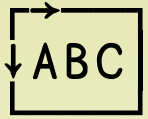
3⍴⊂A



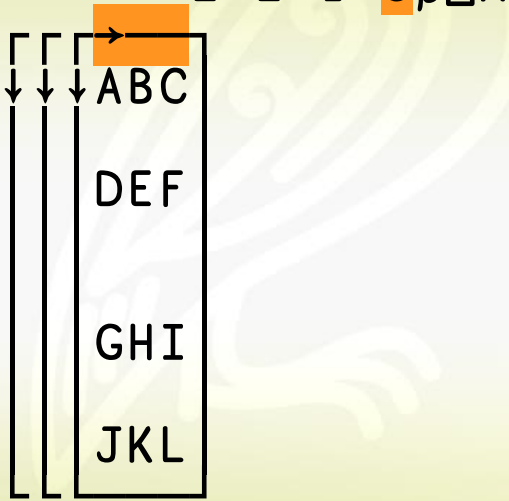
2 1 3⍴⊂A



1 3⍴⊂A

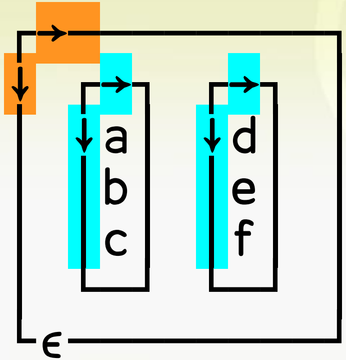


2 2 1 3⍴⊂A

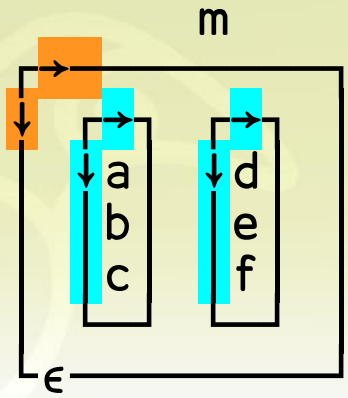


Mixing shapes

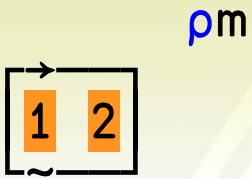
```
m ← ⍉⍪⍪¨'abc' 'def'
m
```

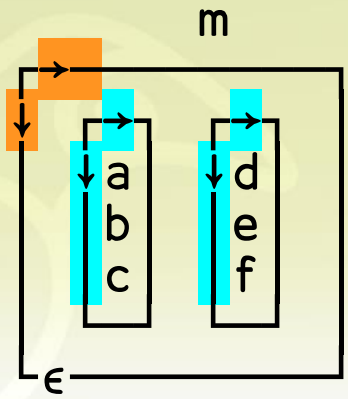


Mixing shapes

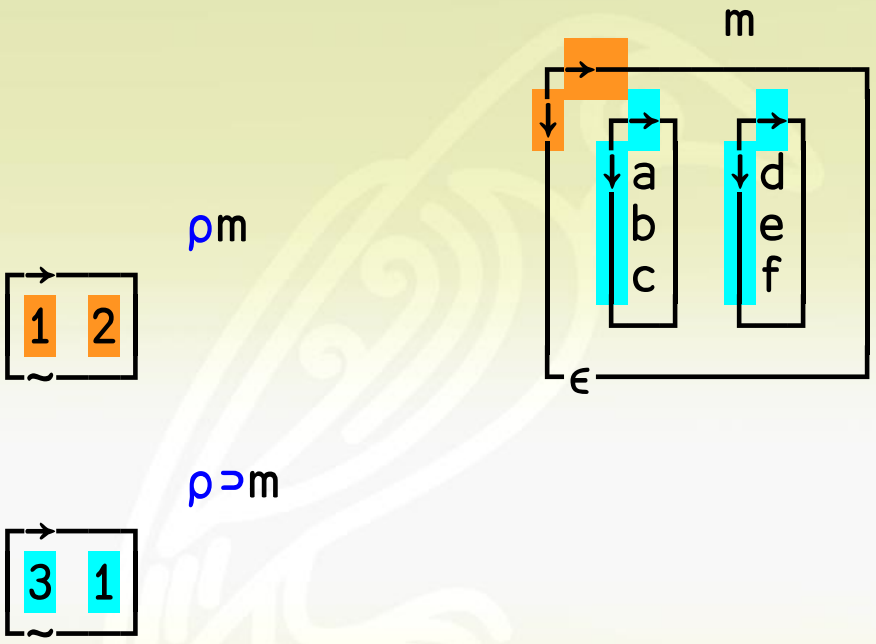


Mixing shapes



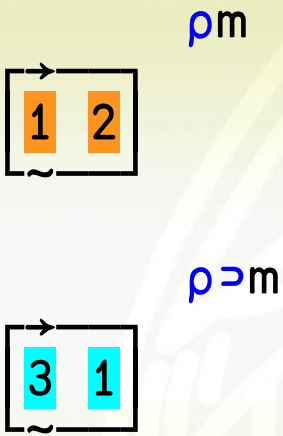


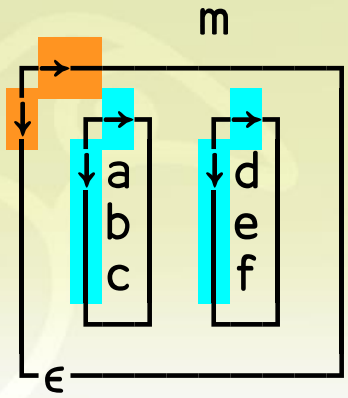
Mixing shapes



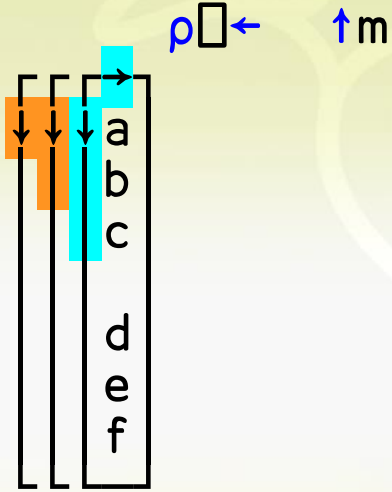
Mixing shapes

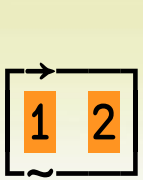
ρ□← ↑m

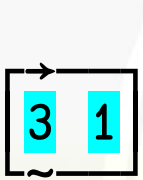


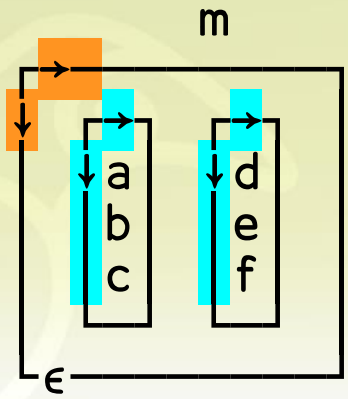


Mixing shapes

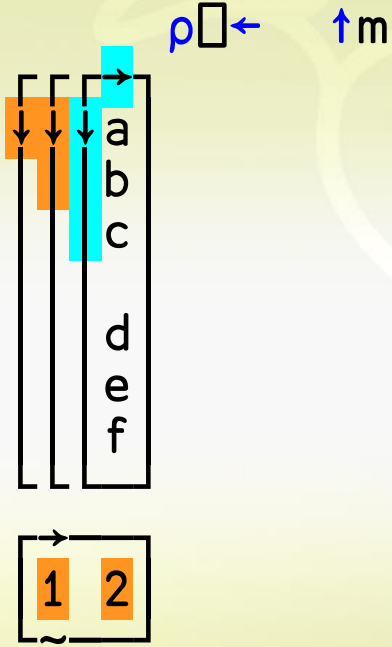


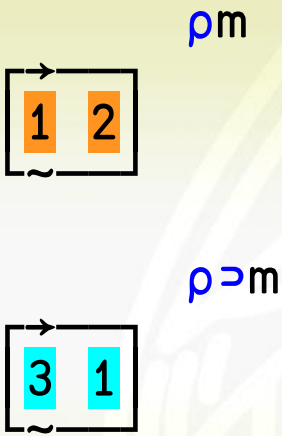


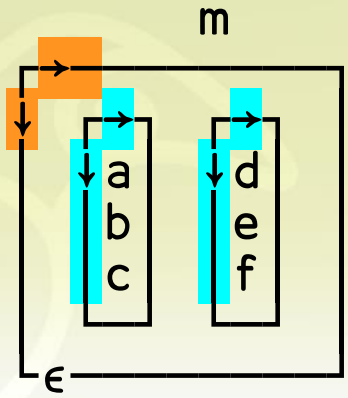




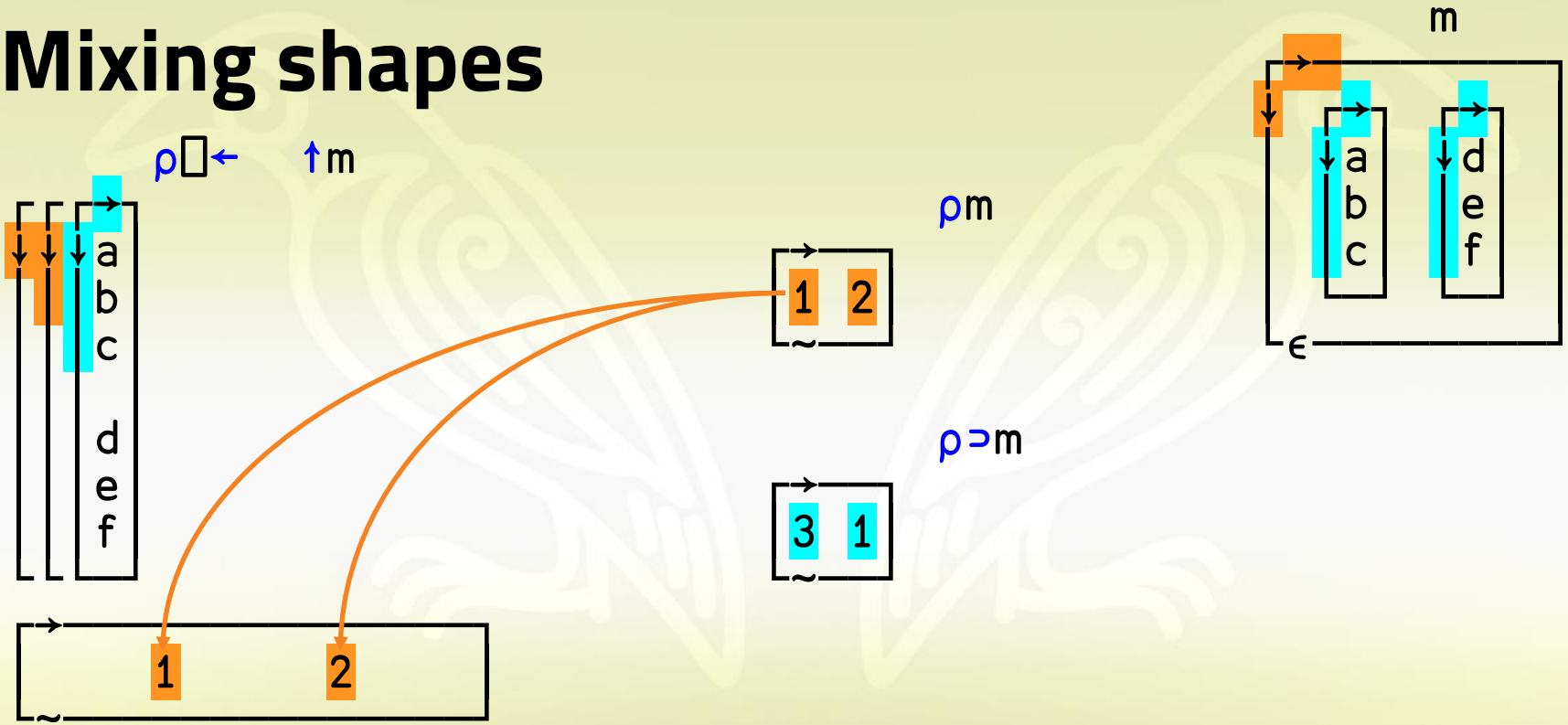
Mixing shapes



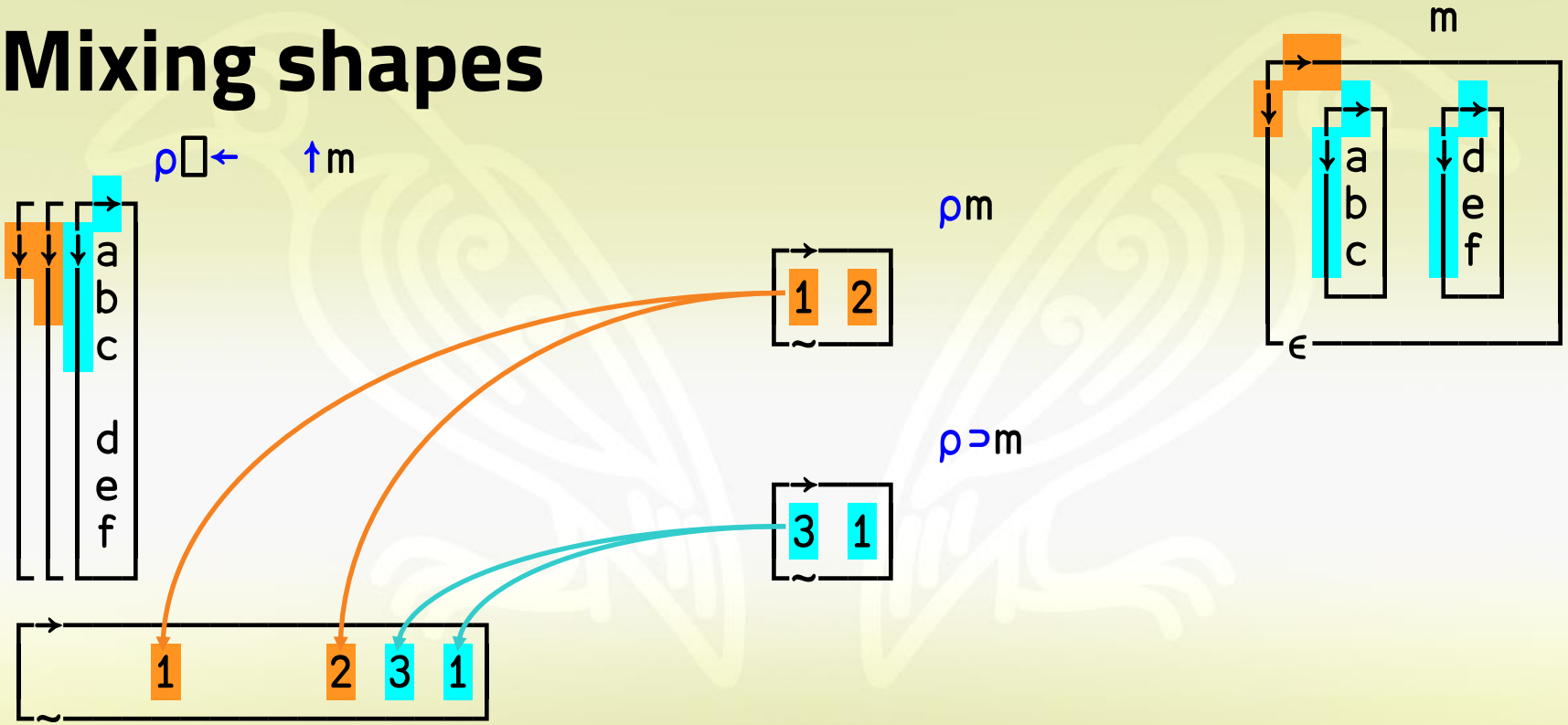




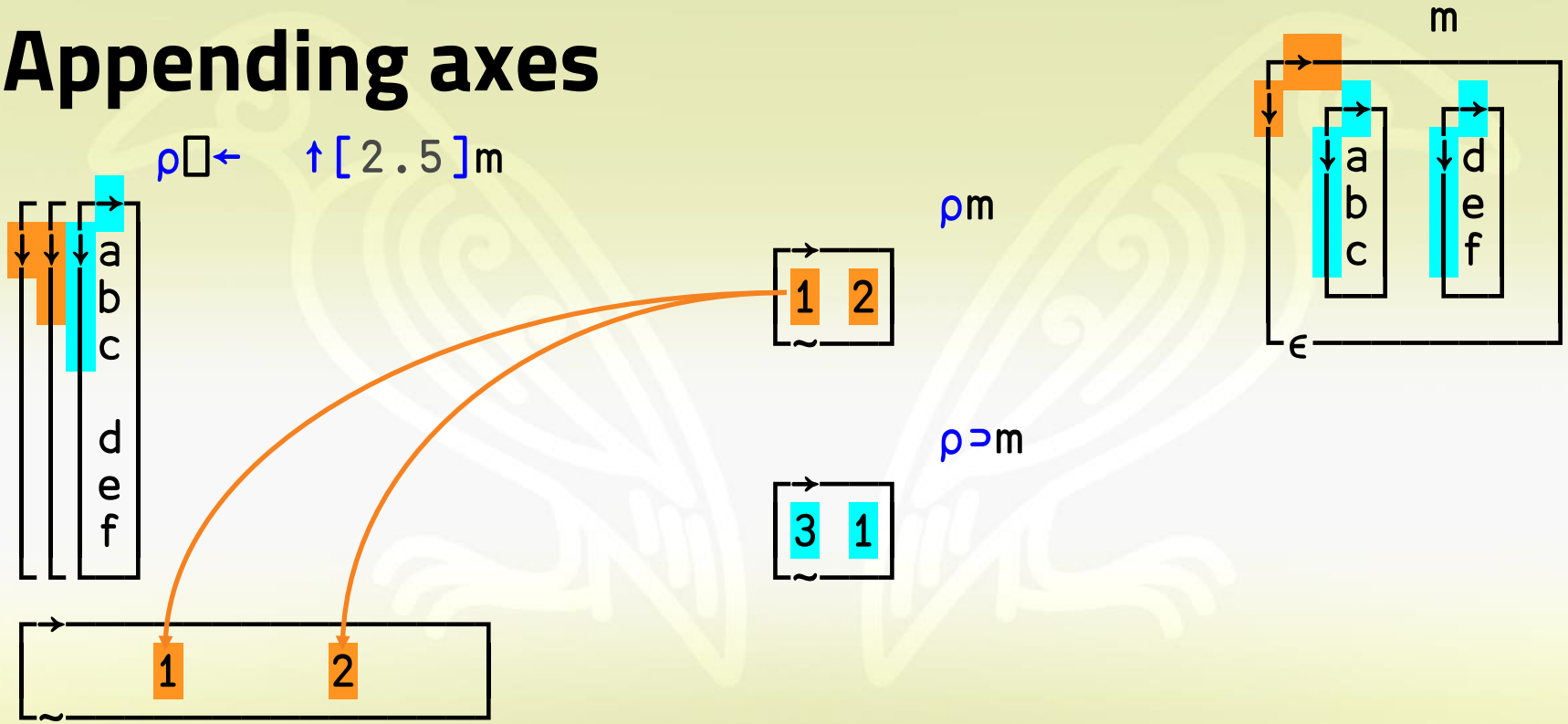
Mixing shapes



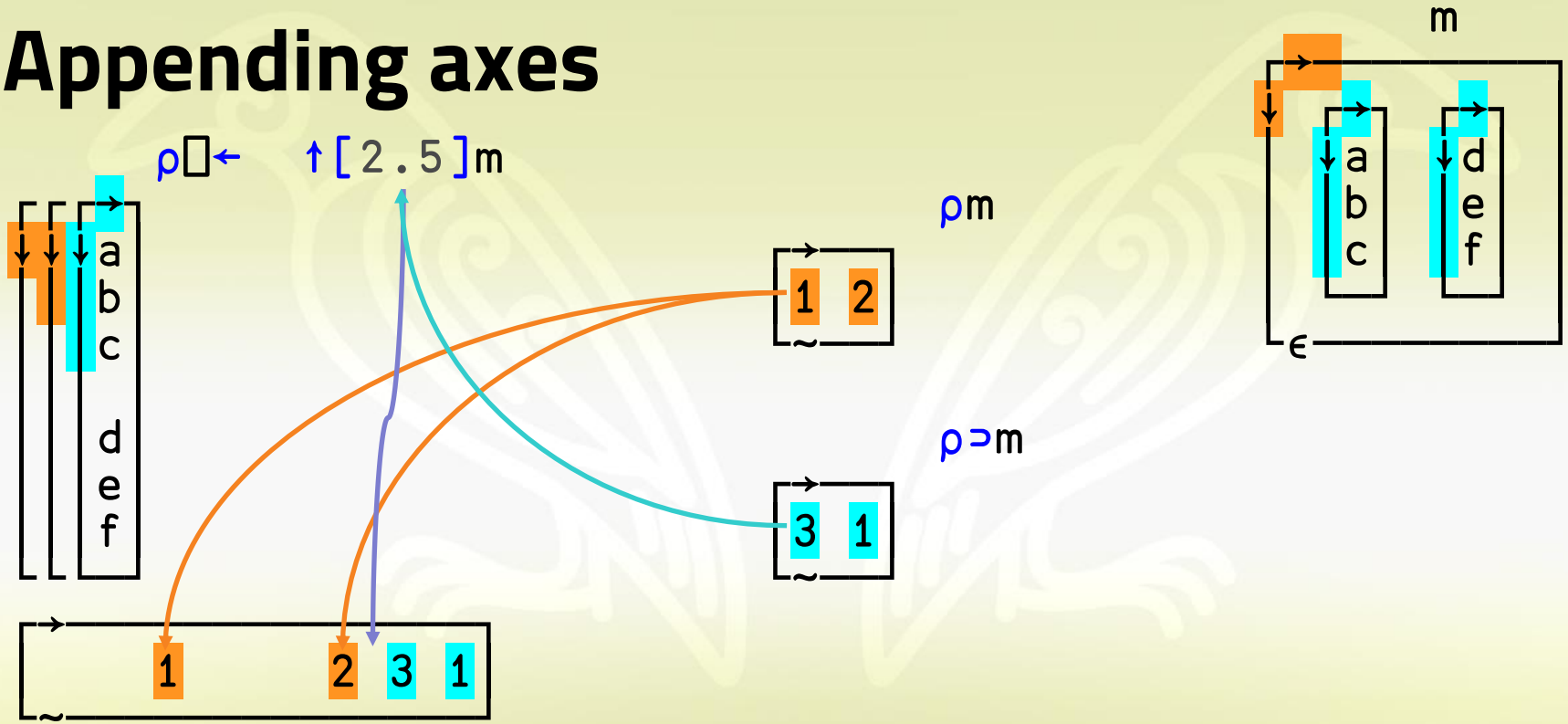
Mixing shapes



Appending axes

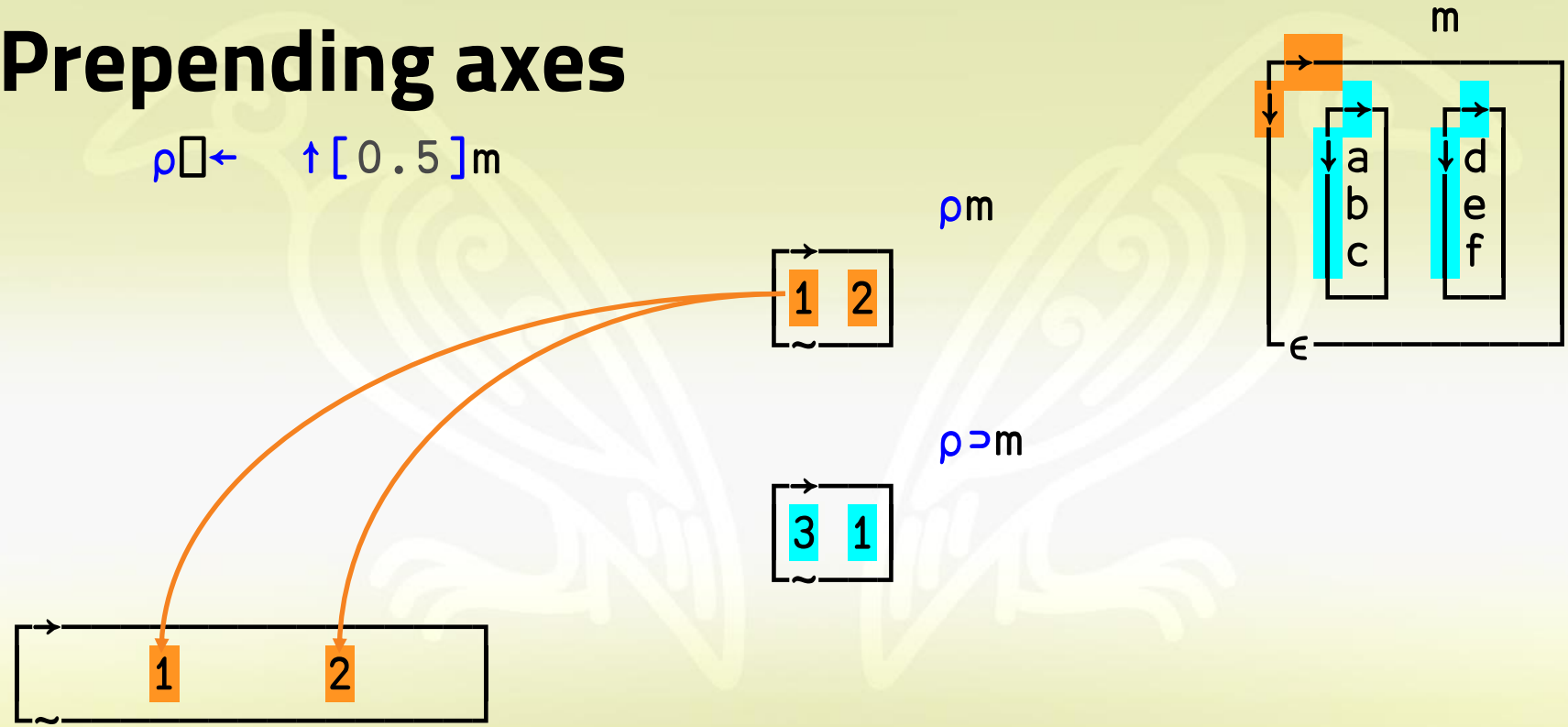


Appending axes

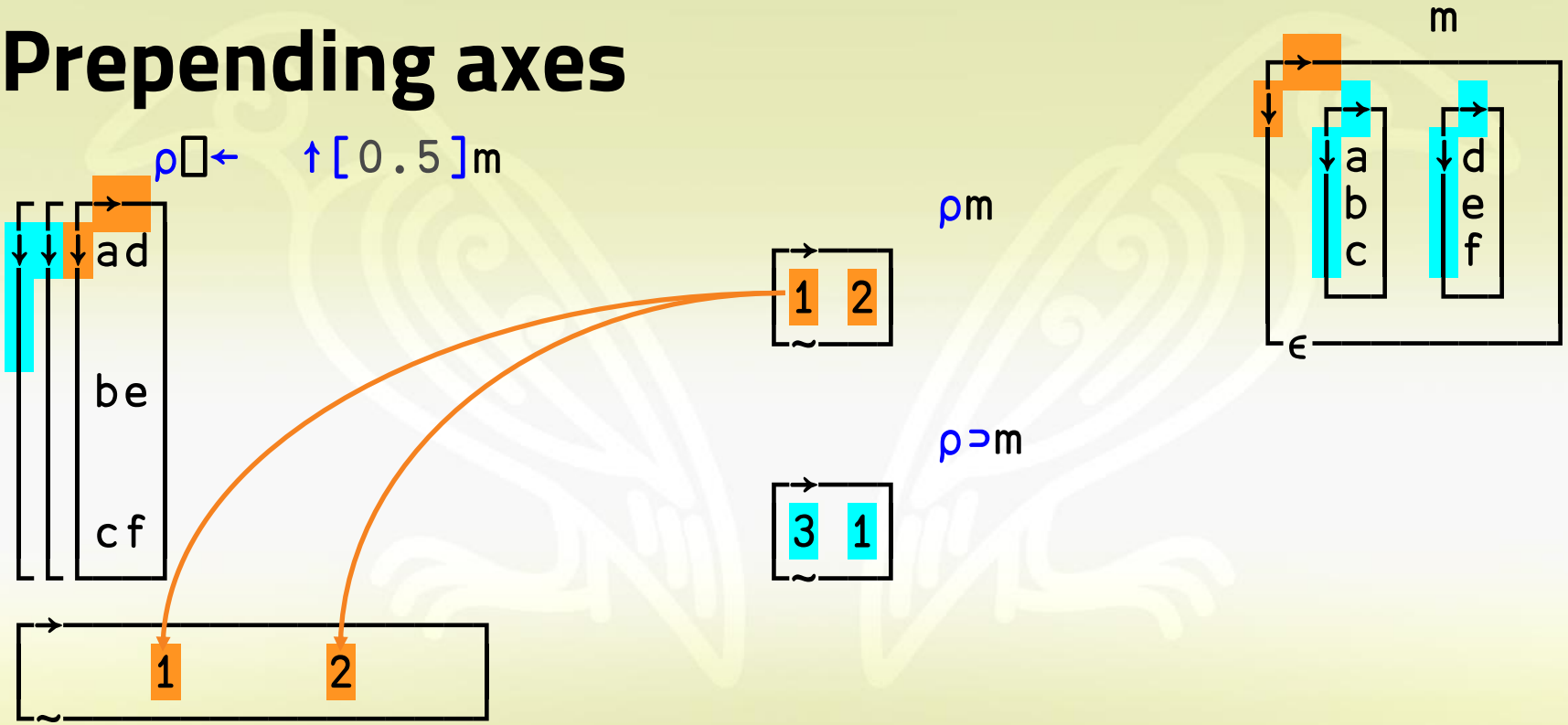


Prepending axes

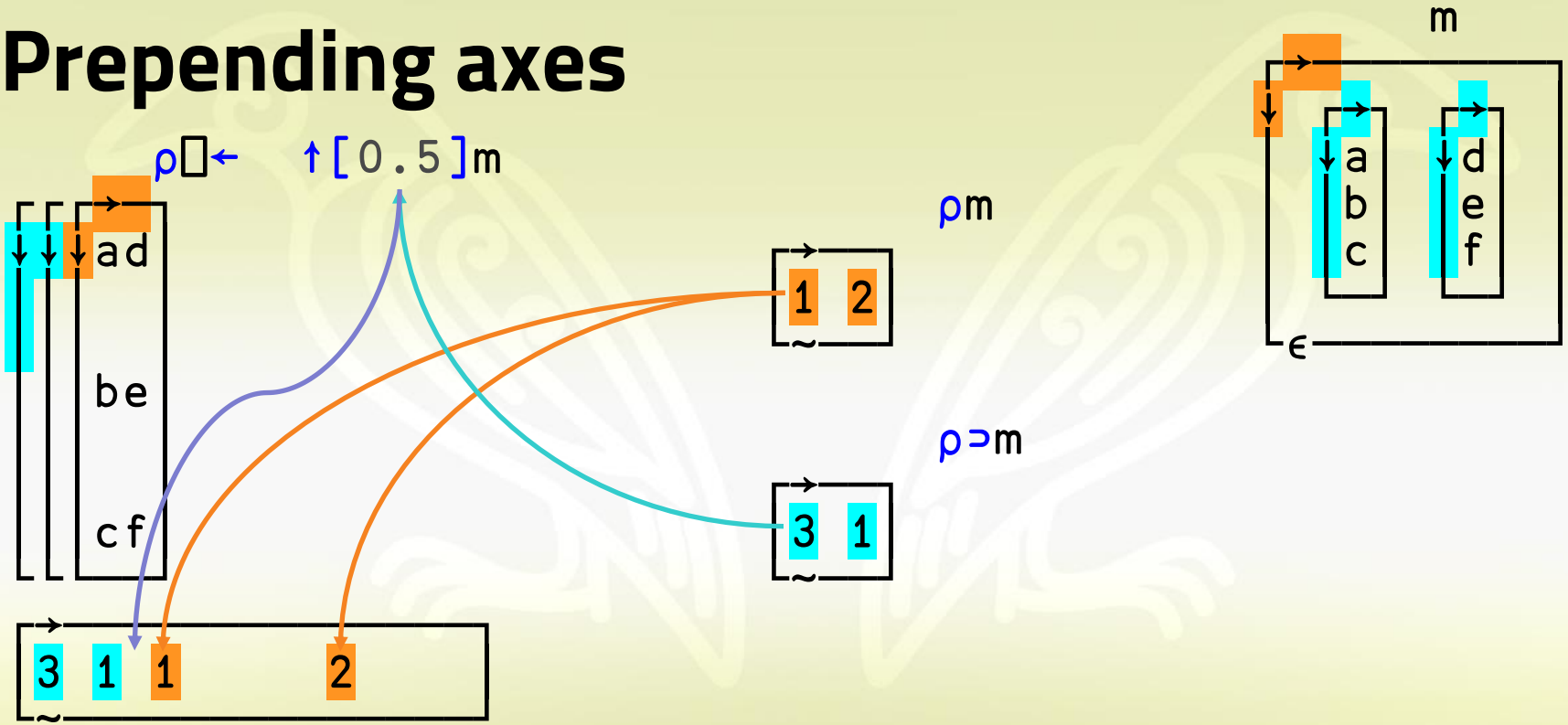
```
ρ⍞←  ↑[0.5]m
```



Prepending axes

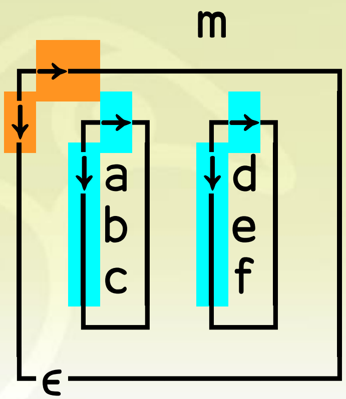


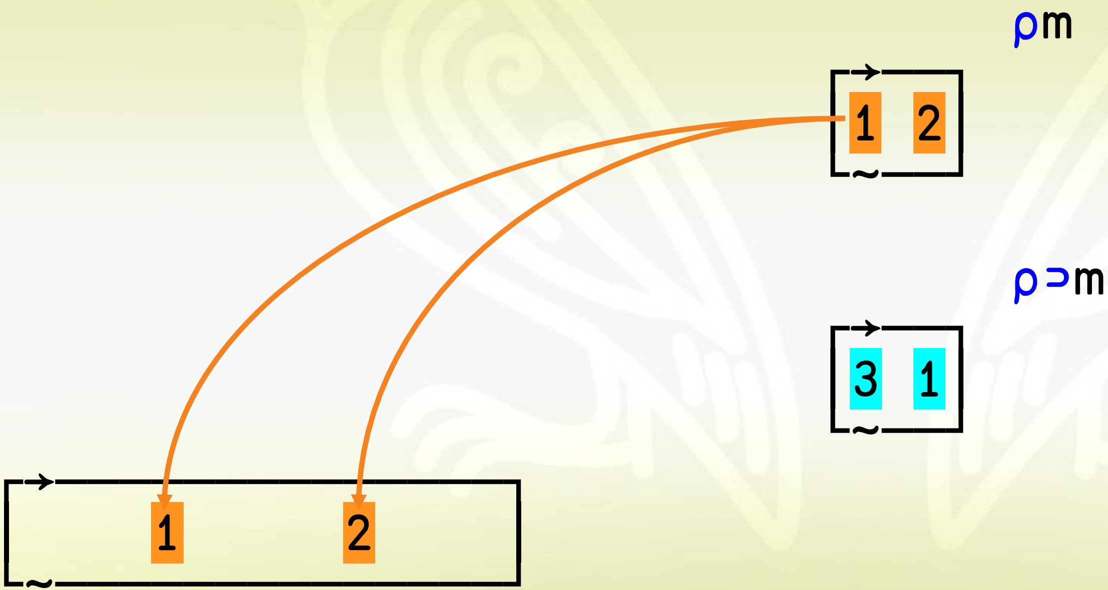
Prepending axes



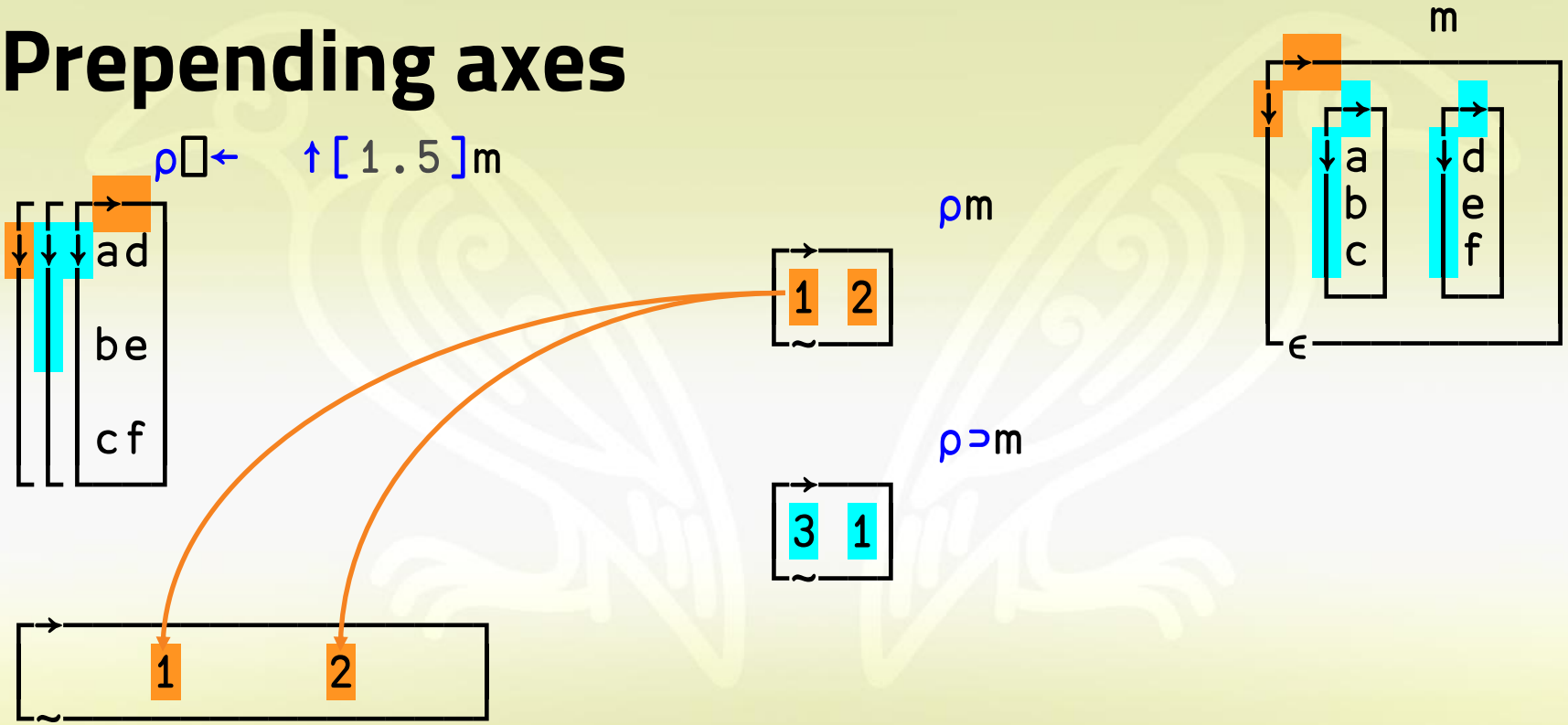
Prepending axes

⍴⎕← ↑[1.5]m

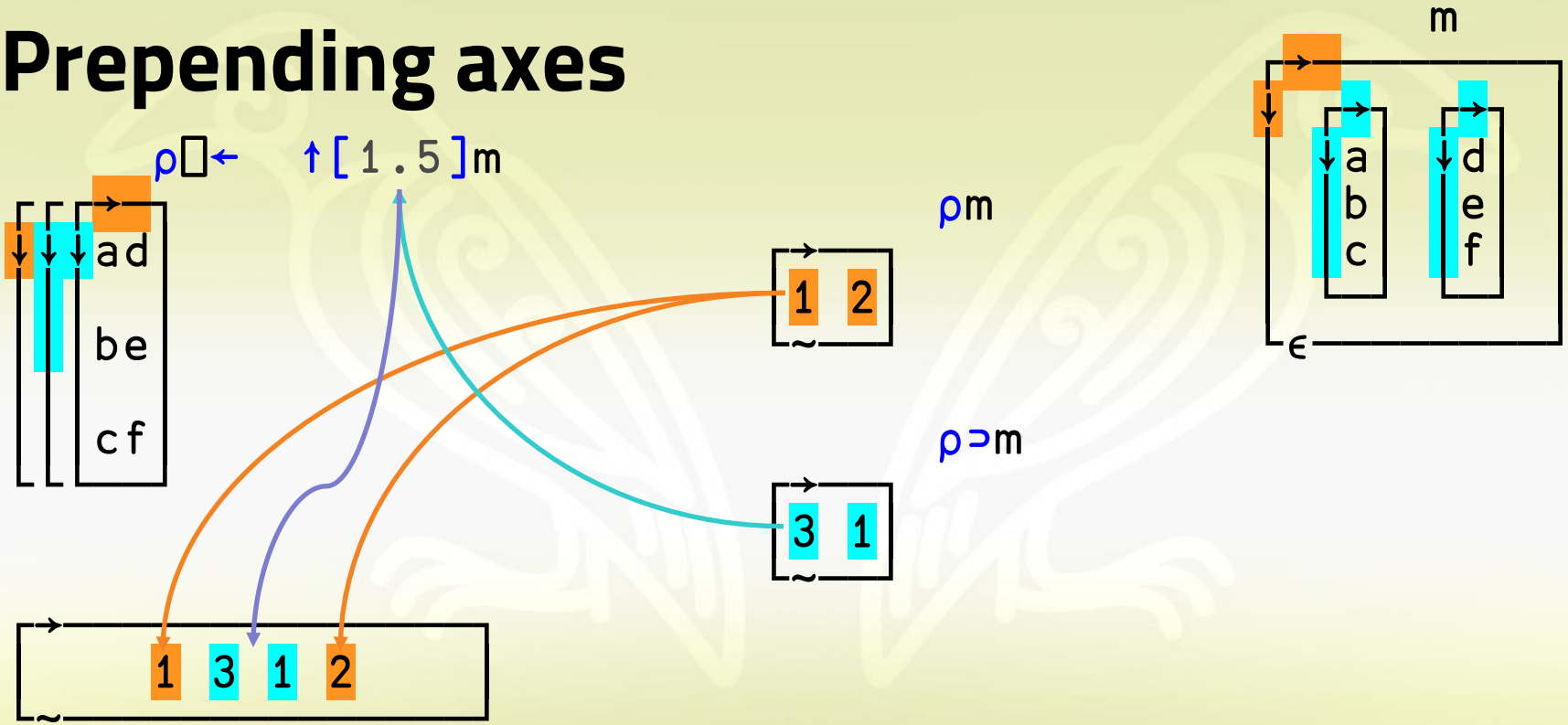




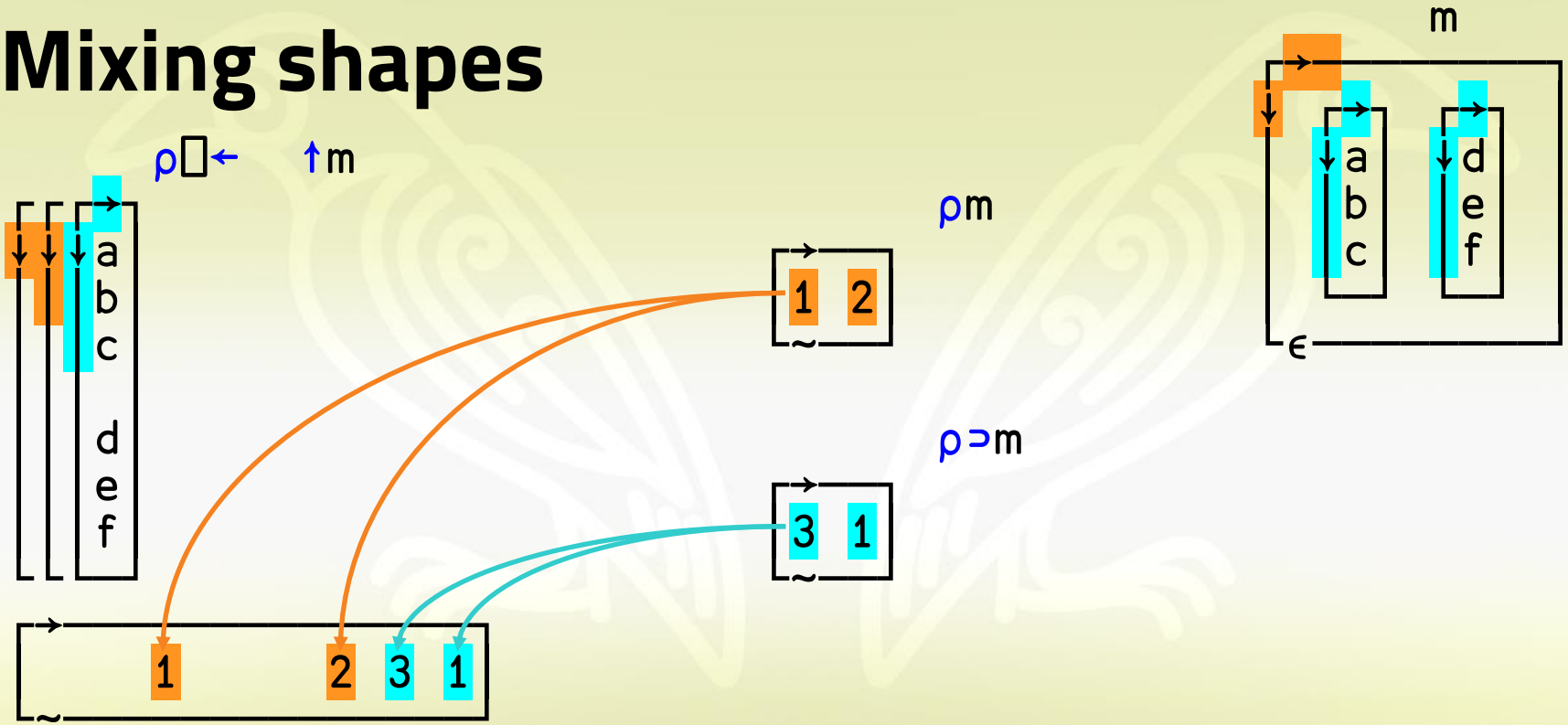
Prepending axes



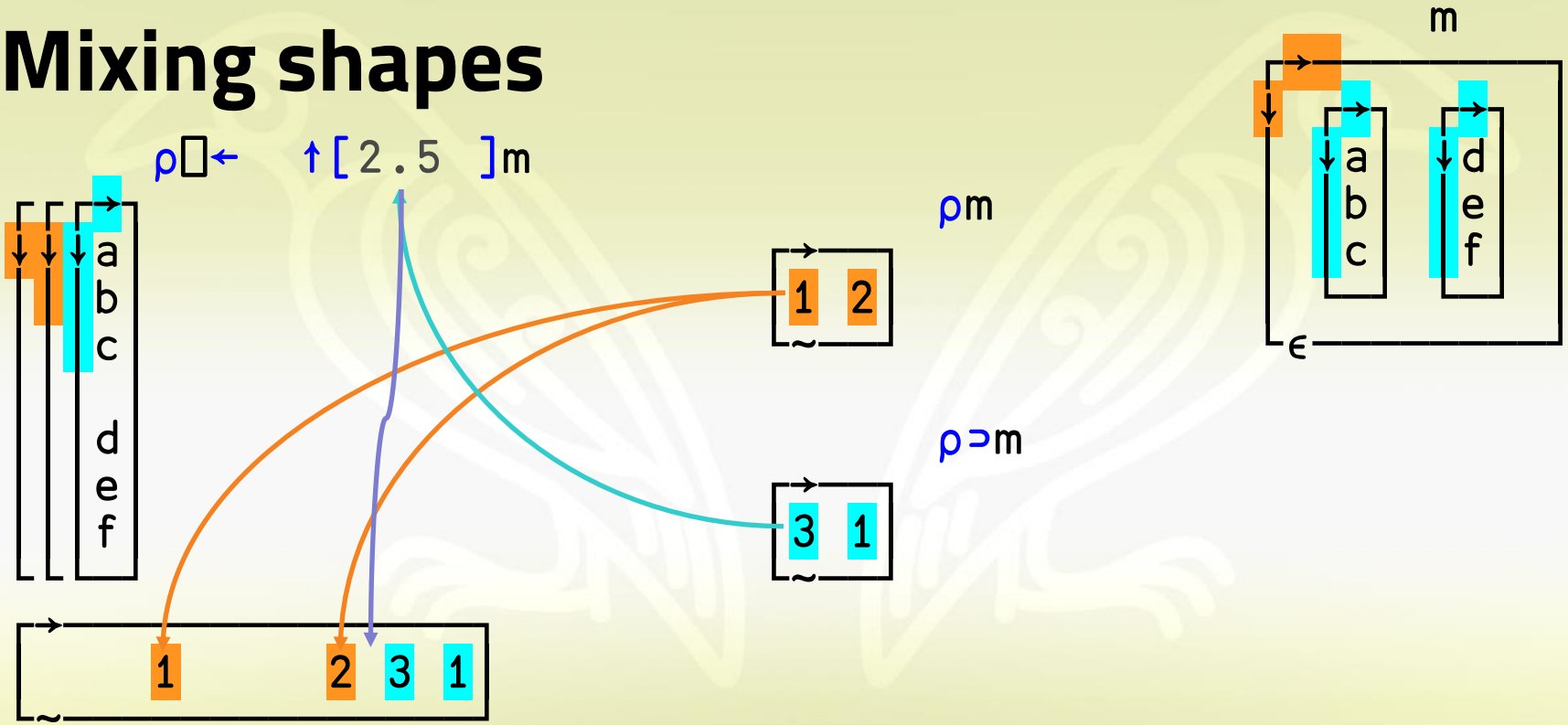
Prepending axes



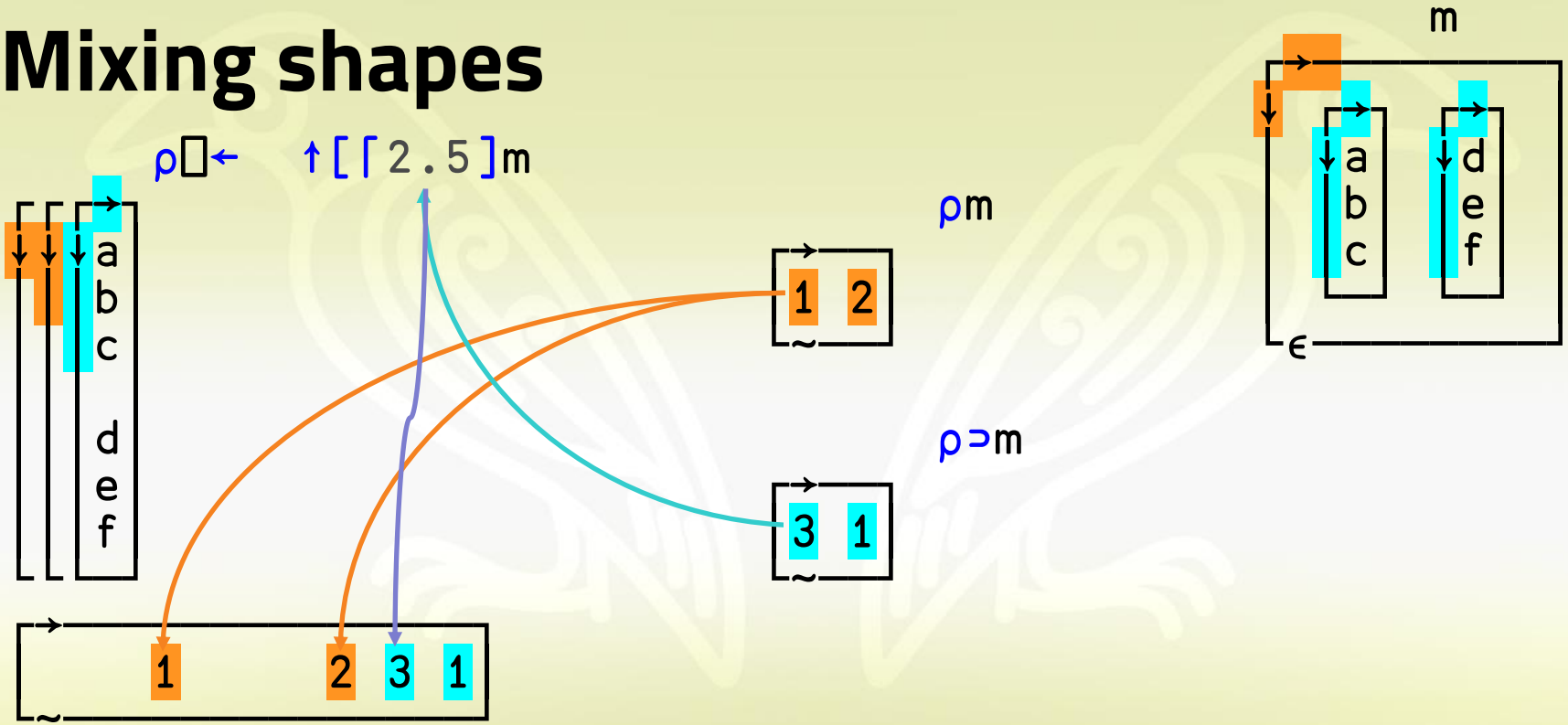
Mixing shapes



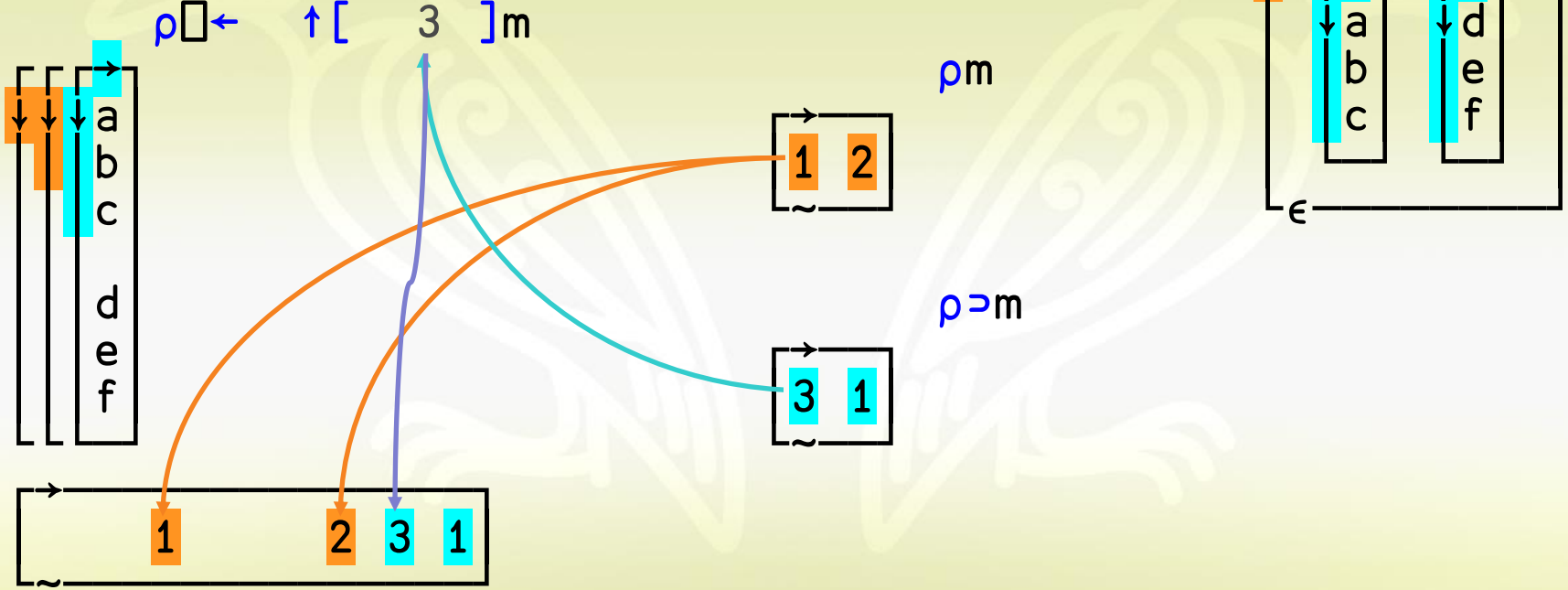
Mixing shapes



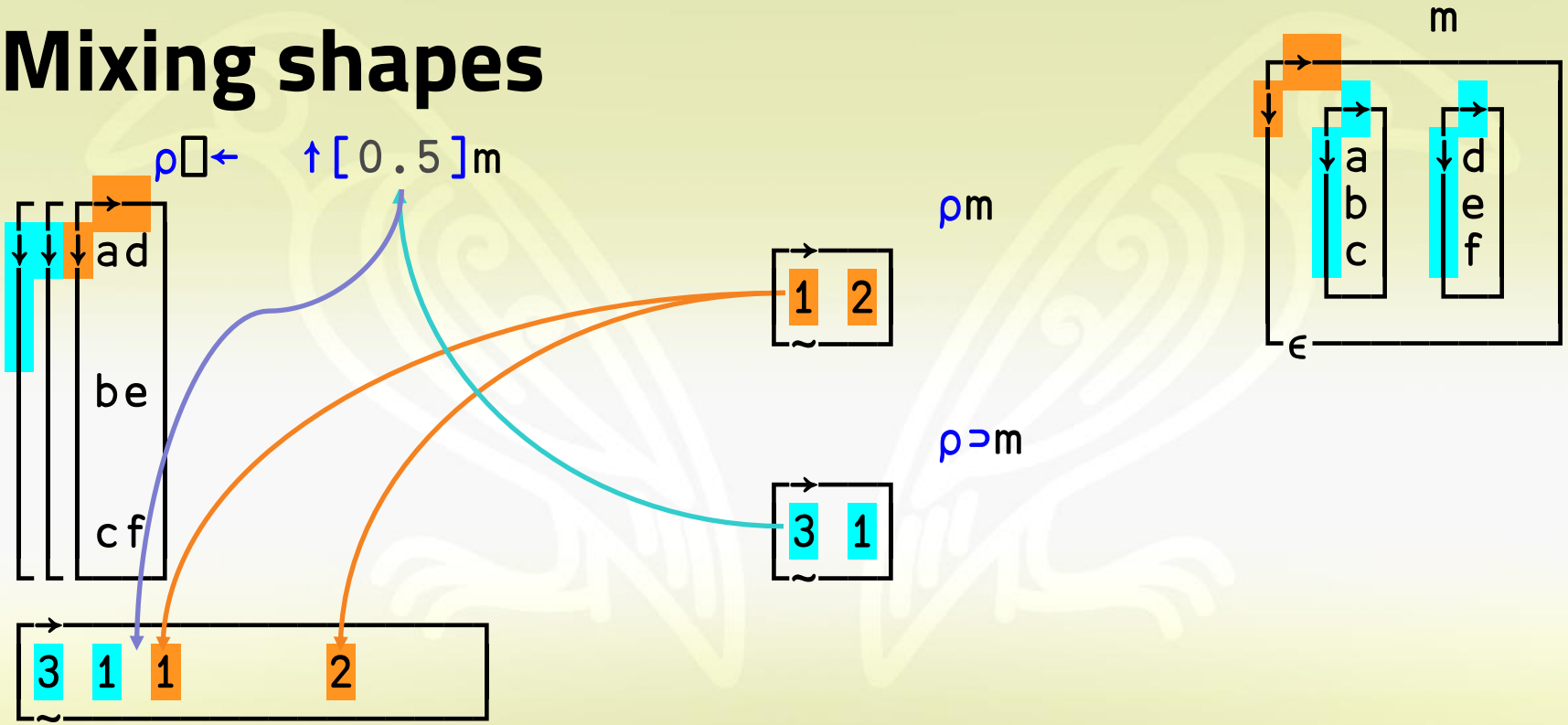
Mixing shapes



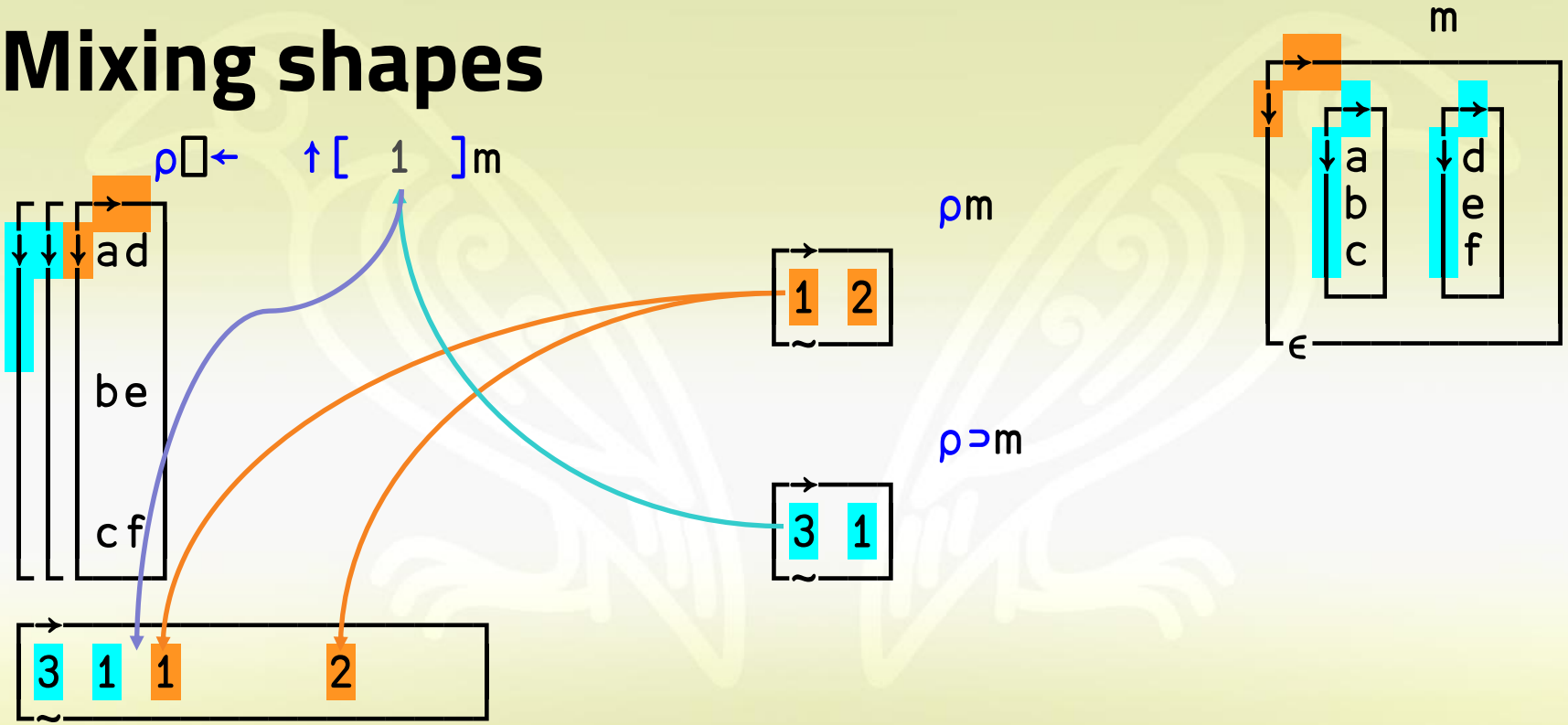
Mixing shapes



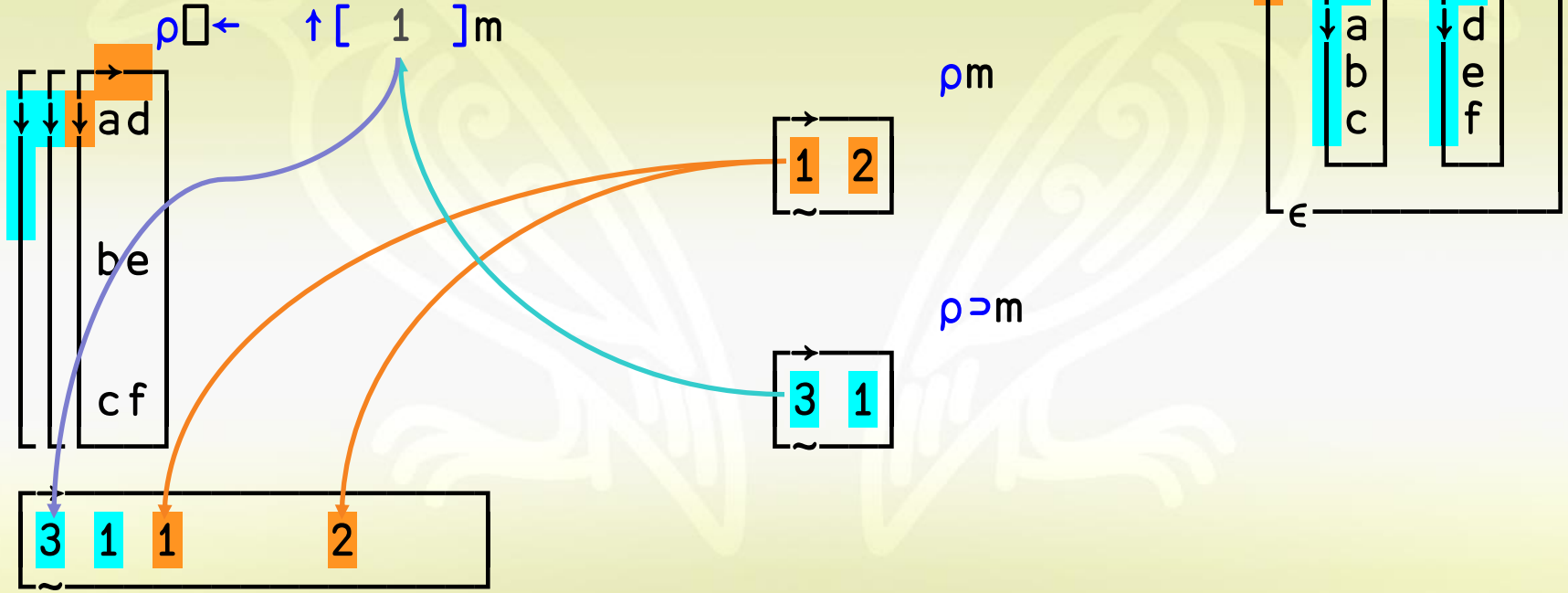
Mixing shapes



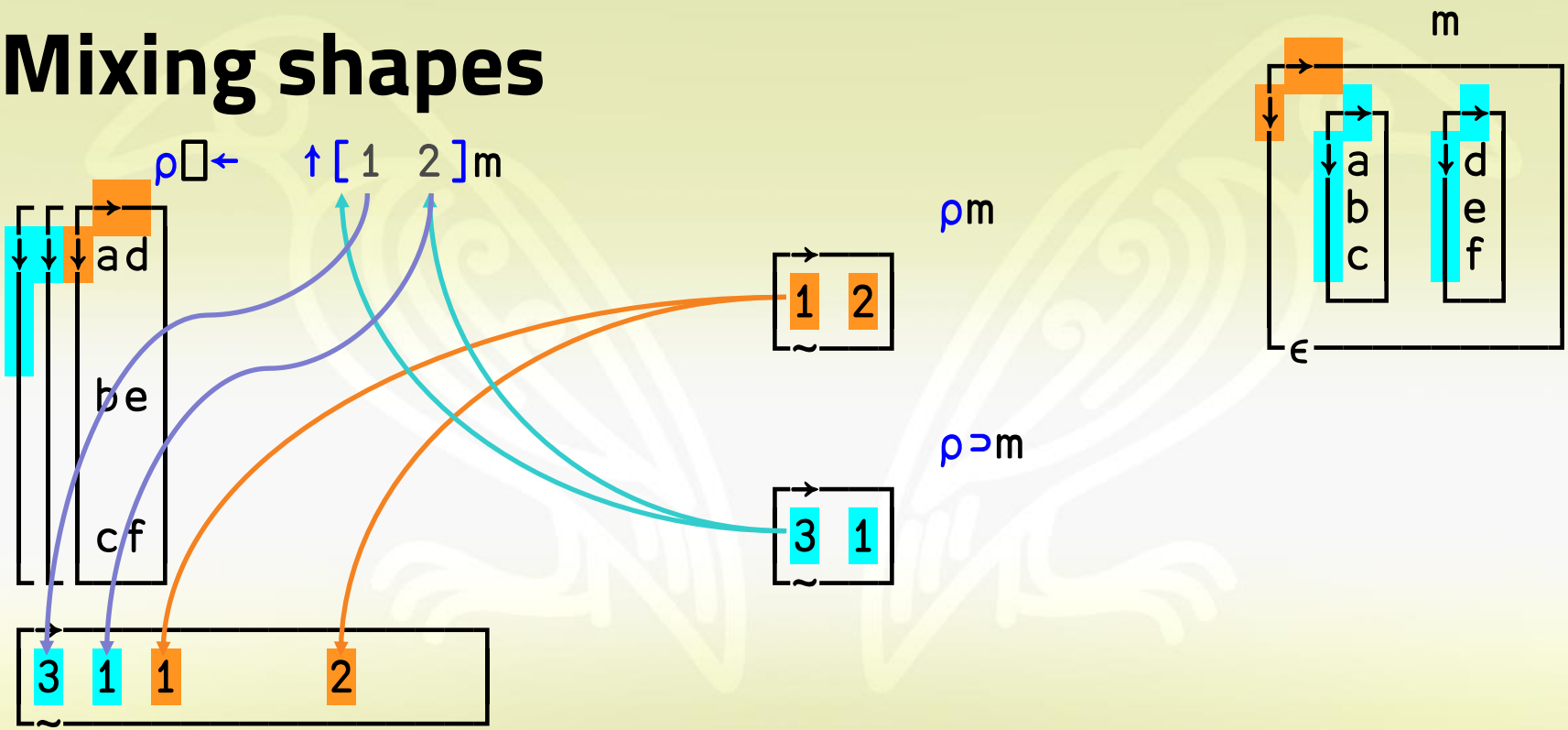
Mixing shapes



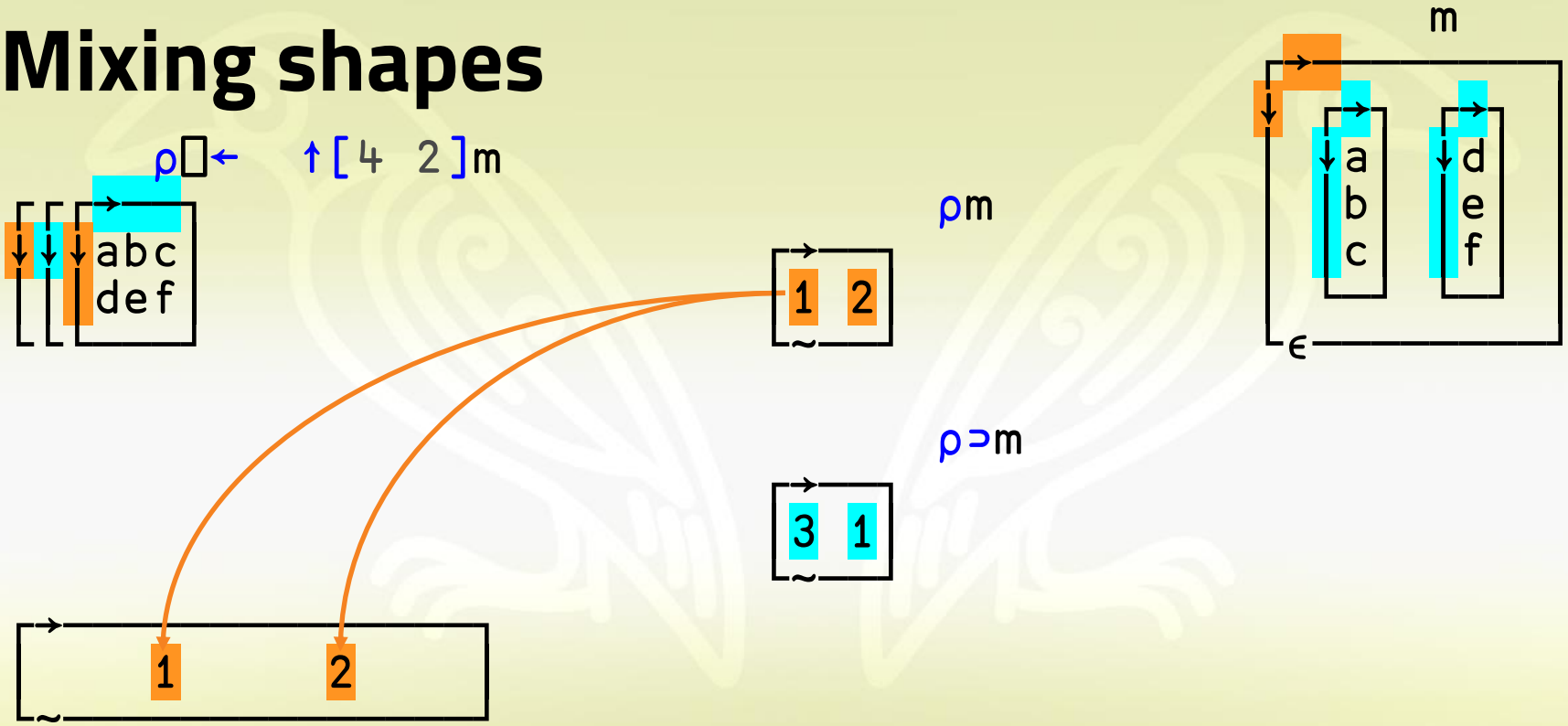
Mixing shapes



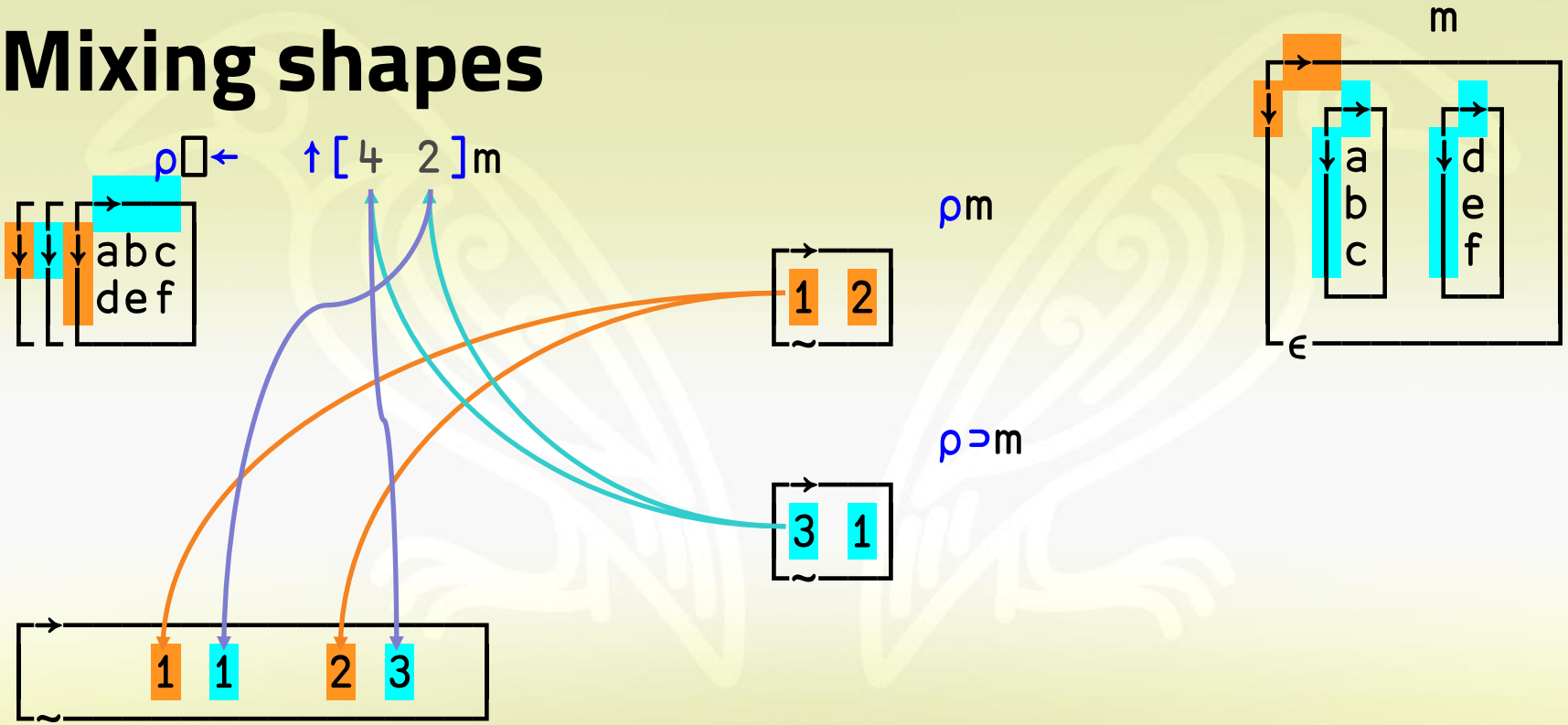
Mixing shapes



Mixing shapes

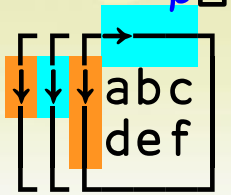


Mixing shapes

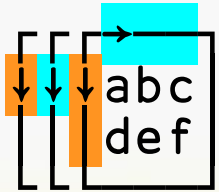


Mixing shapes

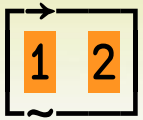
ρ⍬← ↑[4 2]m



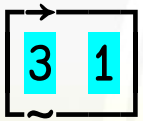
1 3 4 2⍉↑m

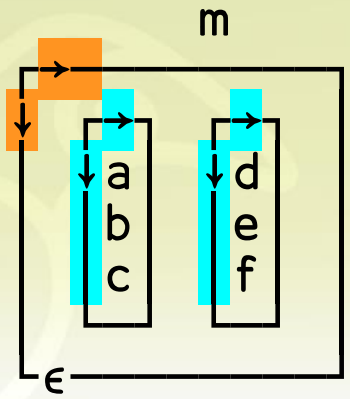


ρm

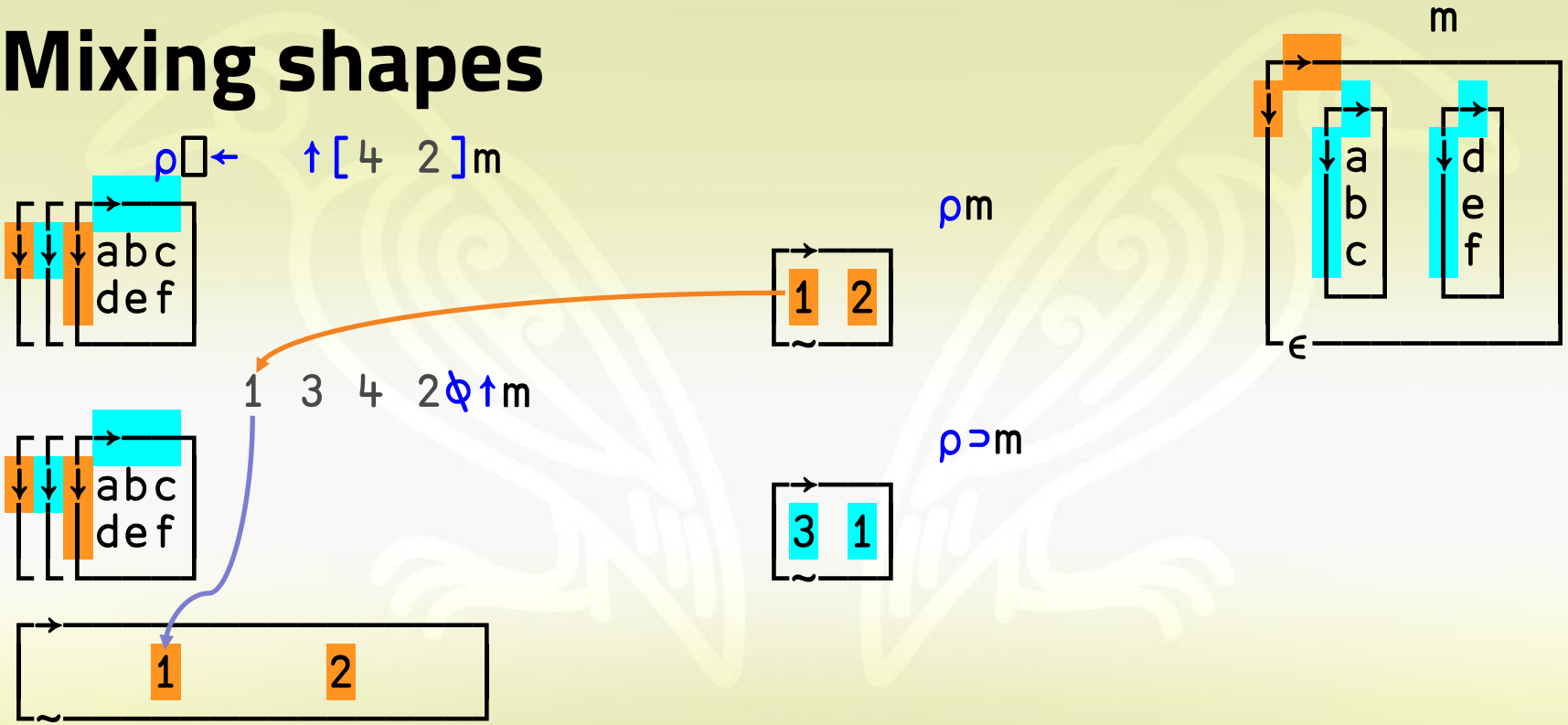


ρ⊃m

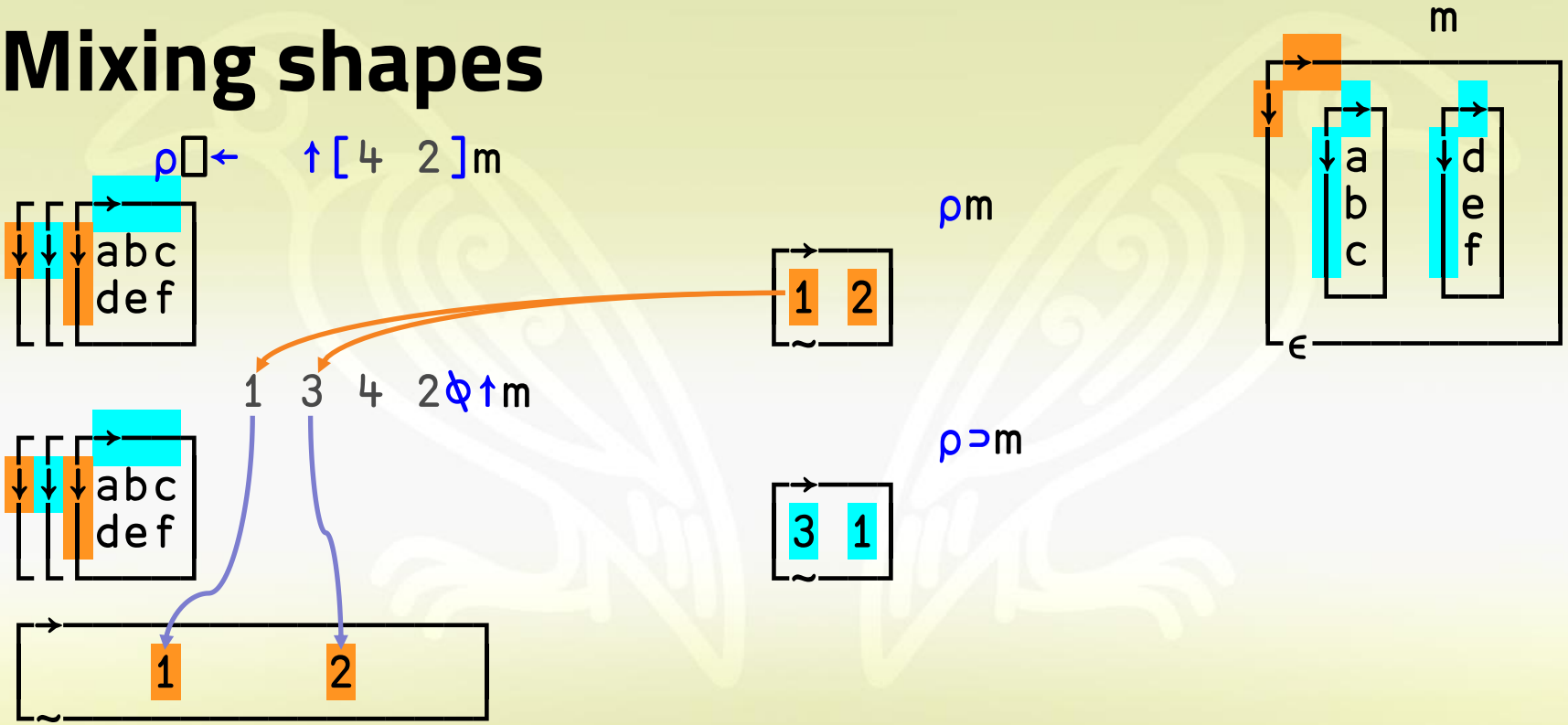




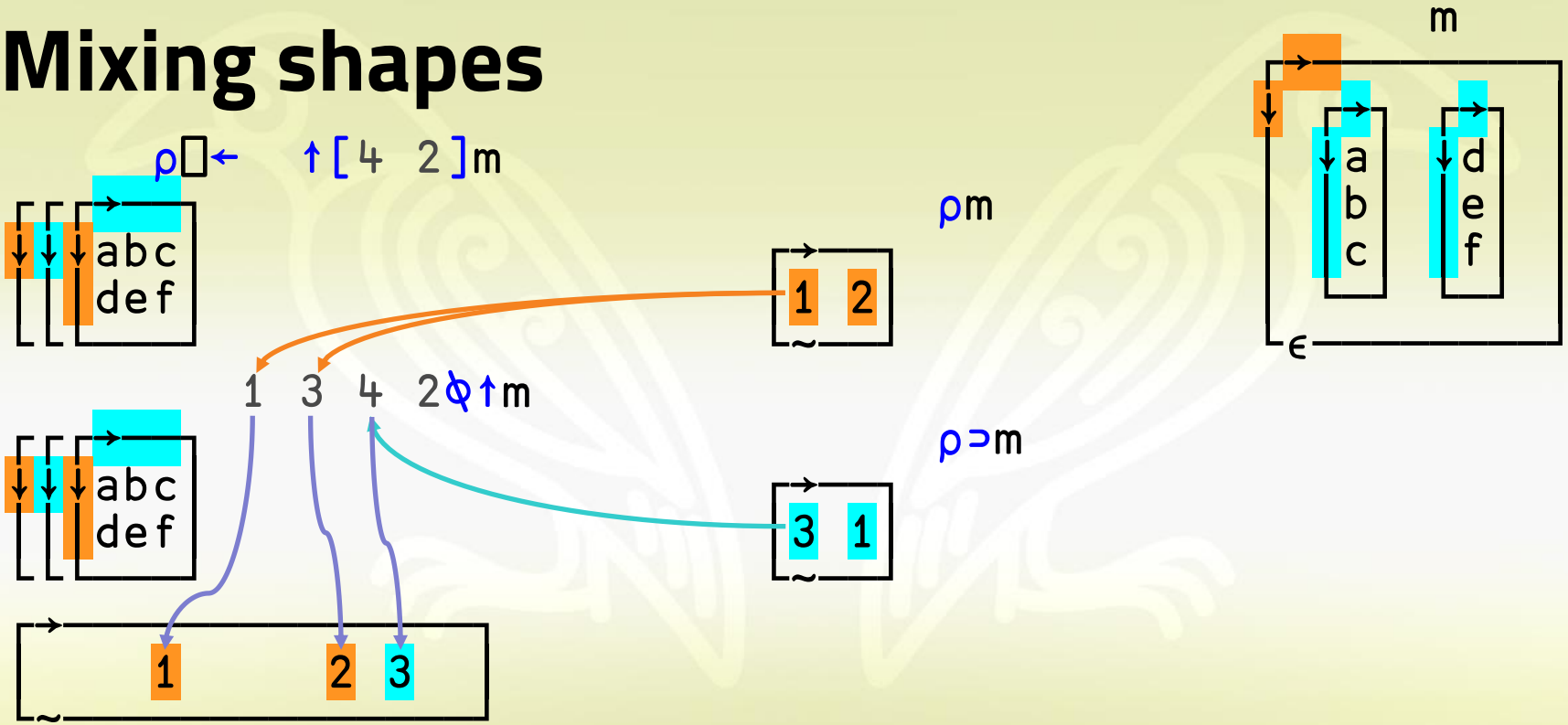
Mixing shapes



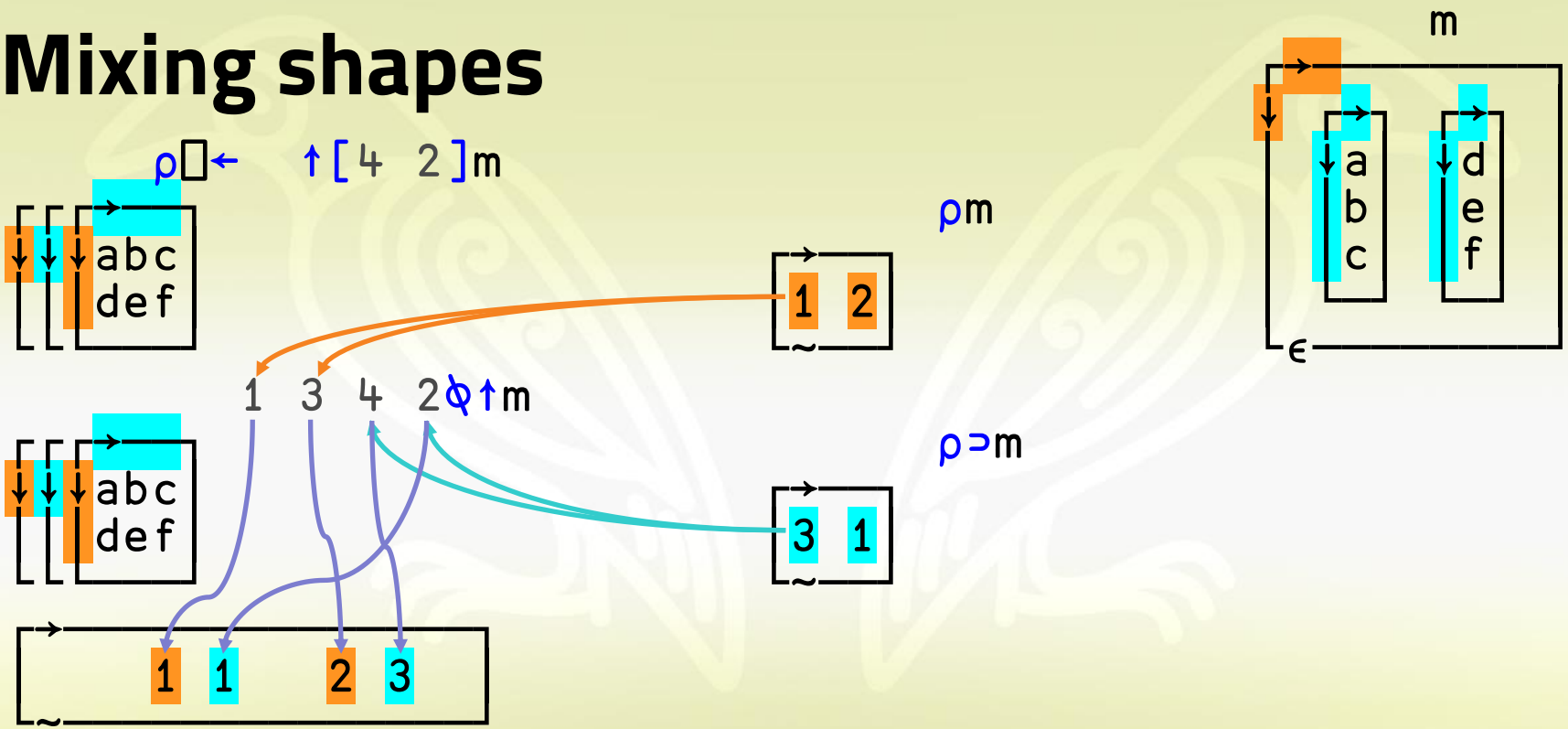
Mixing shapes



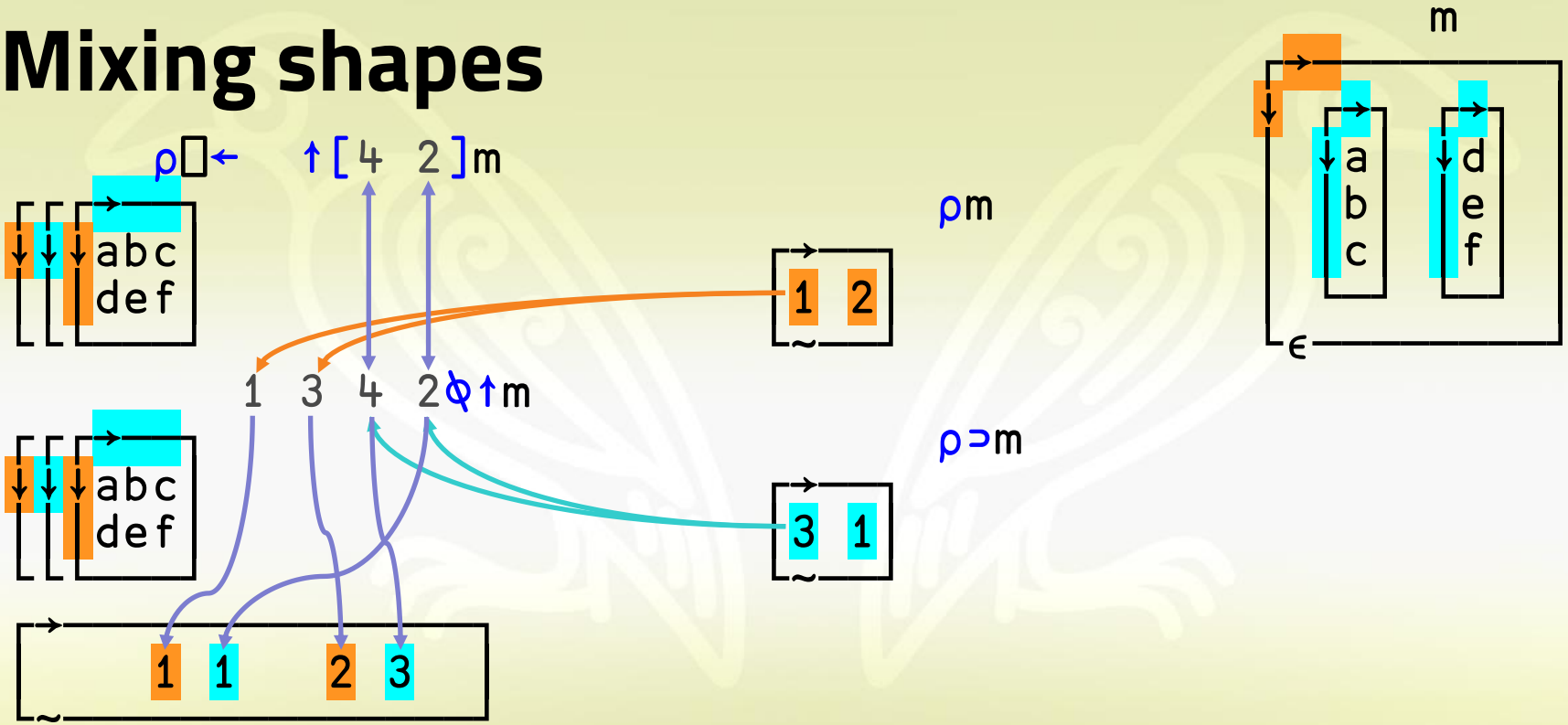
Mixing shapes



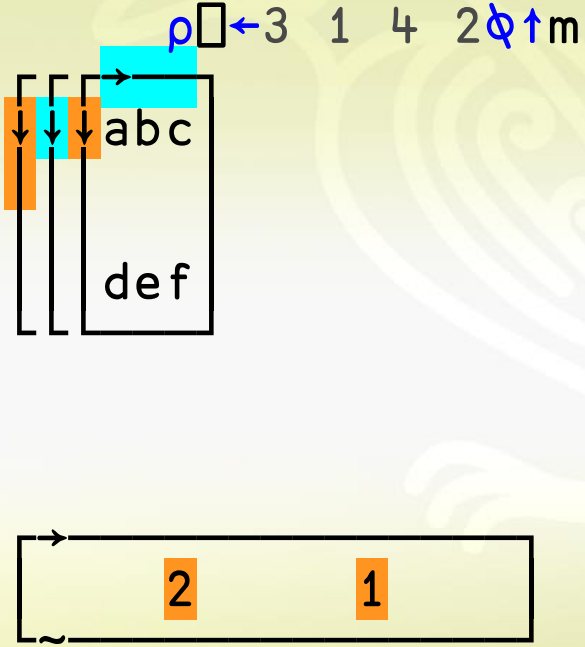
Mixing shapes

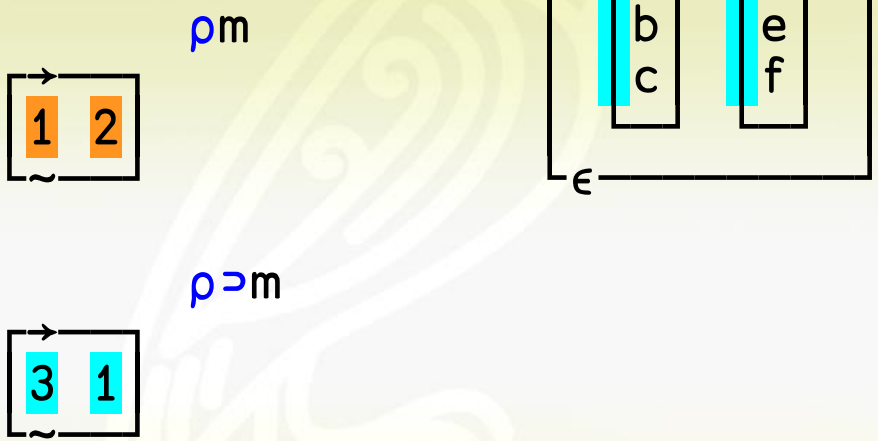


Mixing shapes

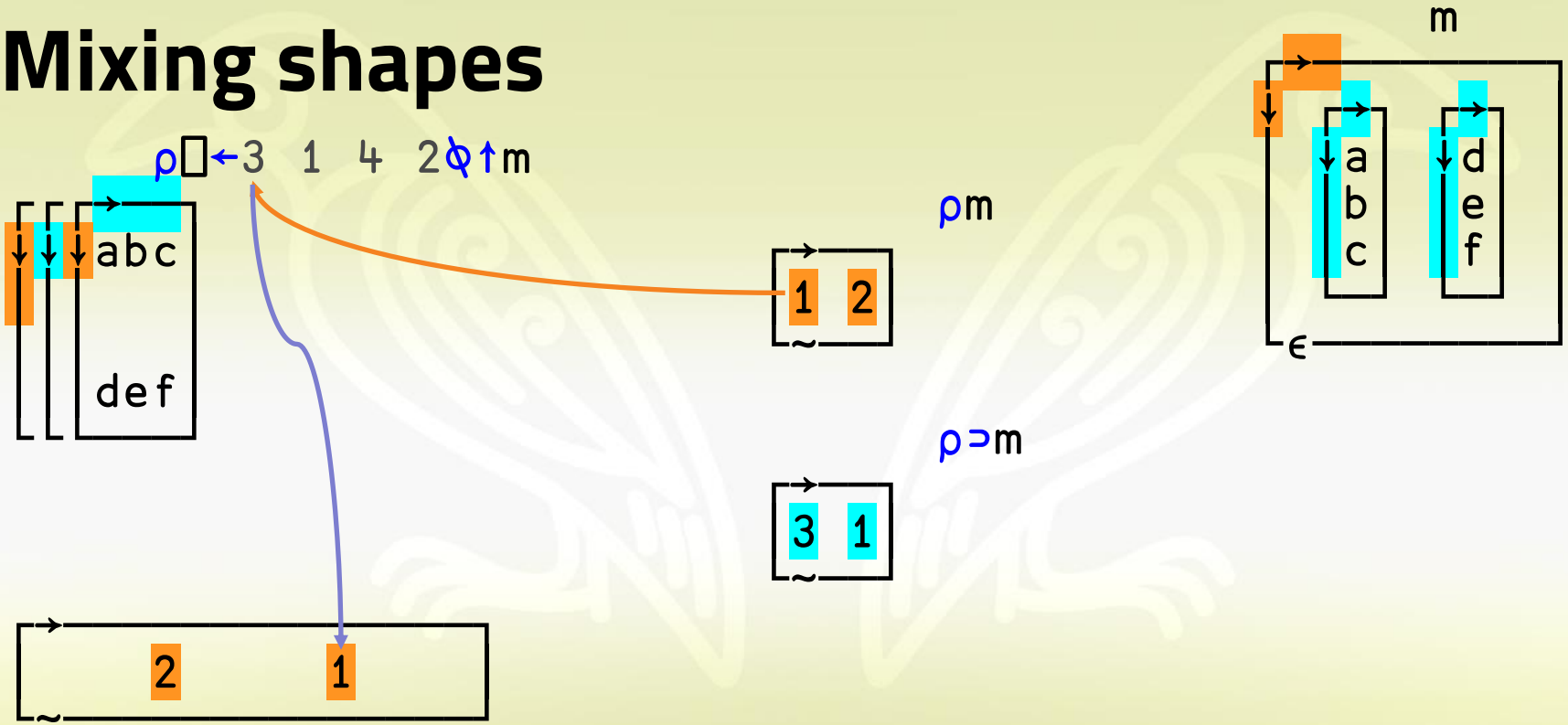


Mixing shapes

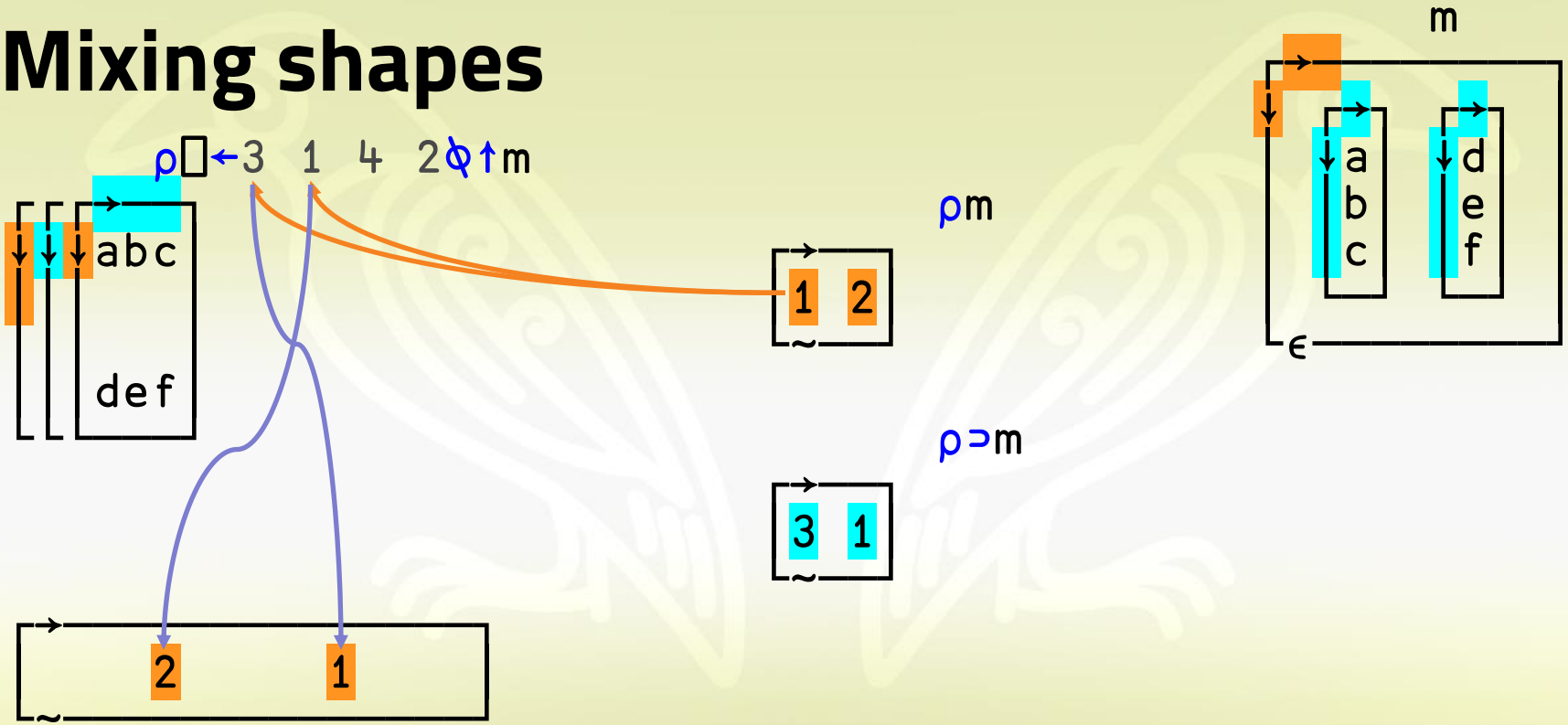




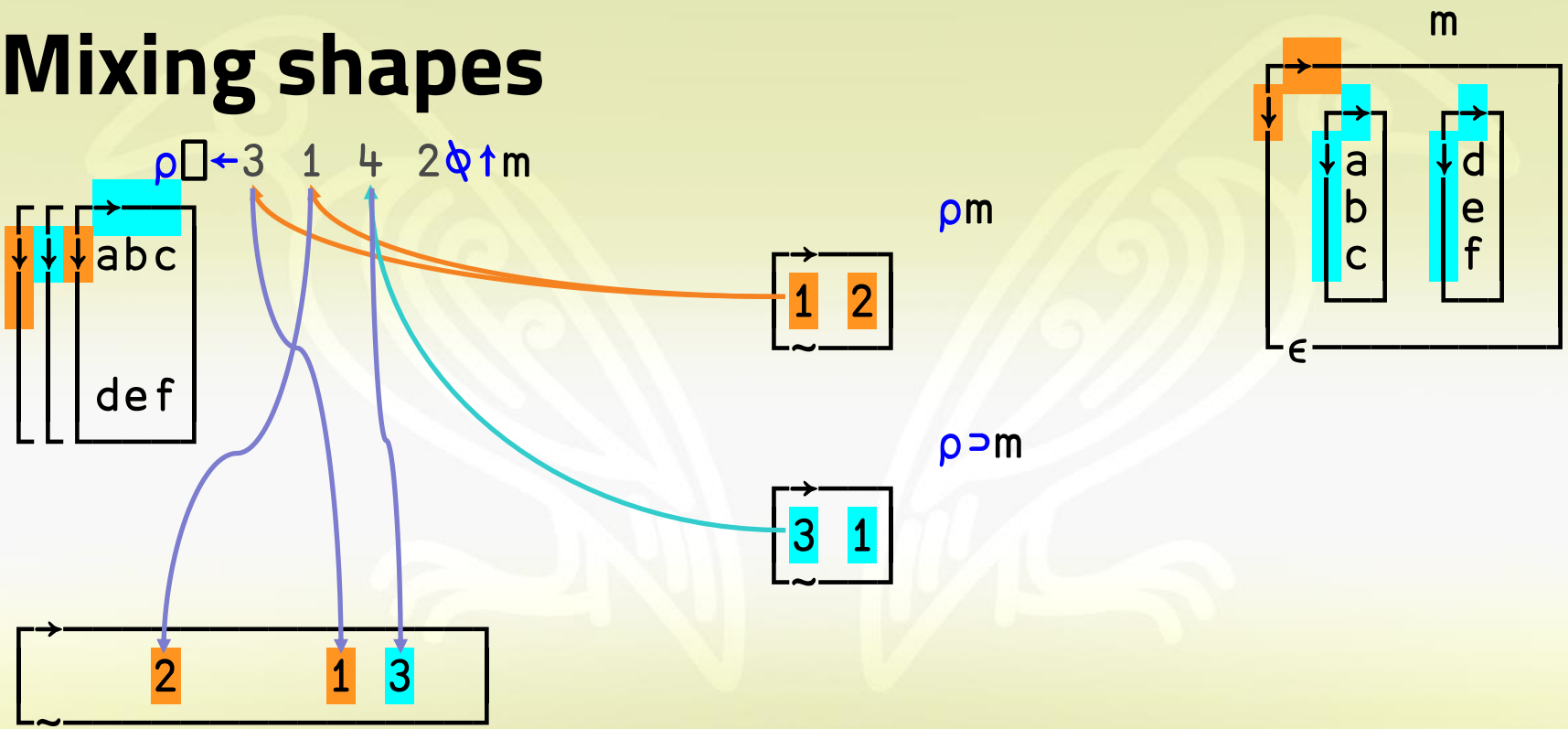
Mixing shapes



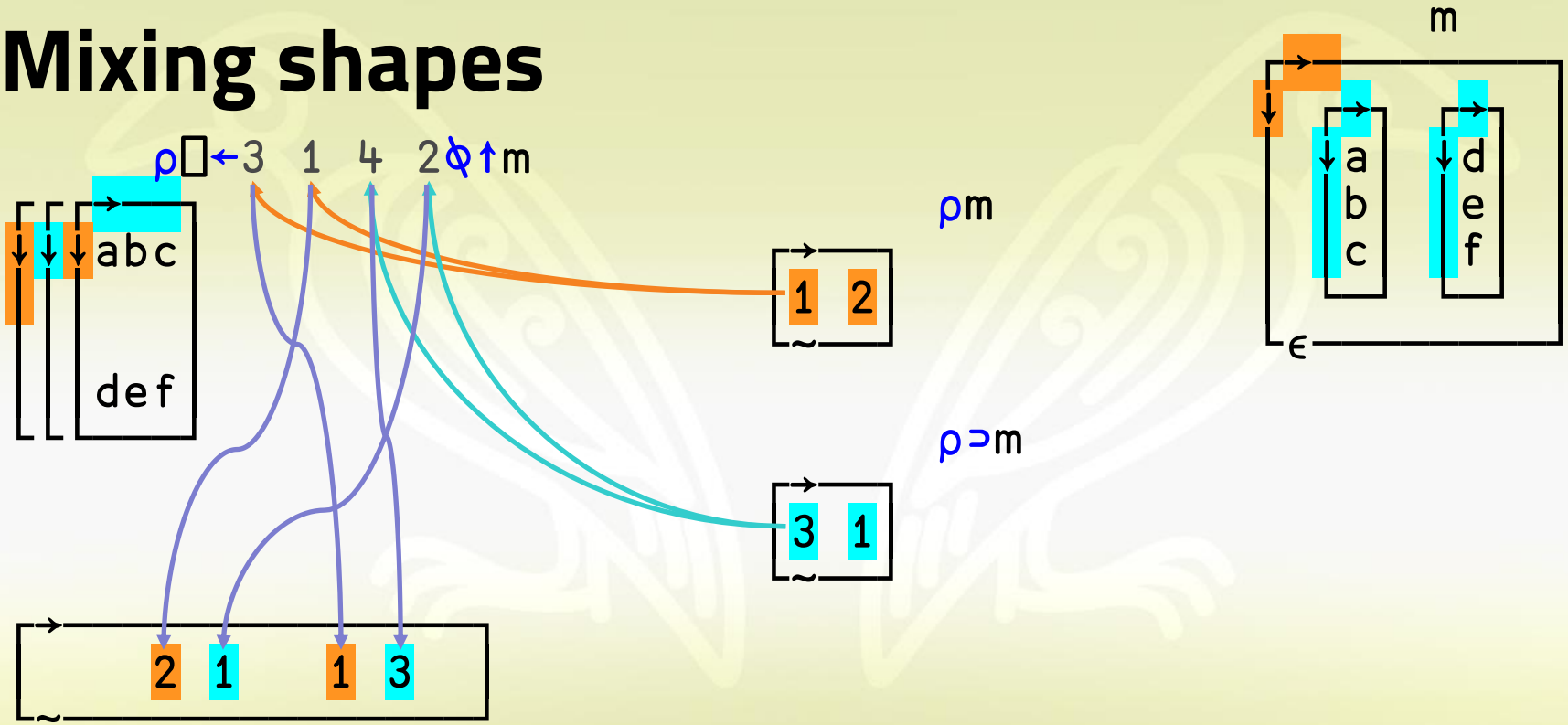
Mixing shapes



Mixing shapes



Mixing shapes



The Rank Operator and Dyadic Transpose

Richard Park

New

⎕C	Case convert
f⍤⍥g	Over
f⍤∘g	Atop
≠Y	Unique mask
A⍨	Constant
⎕DT	Date-time
1200⌶	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Questions?

New

`⎕C`	Case convert
`f⍤⍥g`	Over
`f⍤⍤g`	Atop
`≠Y`	Unique mask
`A⍨`	Constant
`⎕DT`	Date-time
`1200⌶`	Format date-time

Improved

```
⎕JSON⍠'HighRank'
⎕JSON⍠'Dialect'
⎕R/⎕S'\f&'⍠'Regex'
⎕NPUT⍠'NEOL'
⍳Y
X⊂Y
↑[k]Y
```

Thinking in APL: Array-Oriented Solutions

Richard Park