Here we go

1. Good list

With every list element

* 1. For one

type

* + 1. which is

alpha-list.

* 1. And
     1. Back

Again

1. To L1

Test 1: BL3 / p > BL1p conditional (pass, with extract)

* Conditional pass
* BL2

Bl3

More Bl3p, Bl3p >. Bl3p test (should pass)

Interceding extract

Target: Bl1p, should be ok

Test 2: BL3 / p > BL1p conditional (Not OK, with extract)

UL1

UL1p

* BL2

Bl3

Interceding extract

Target: BL1p, fail test2 list\_nesting

Test 3: BL1 / p > NL1p Not OK

* BL1

NL1p should fail test3 list\_nesting

Test 4: BL2 / p > NL3p Not OK

* BL1
* BL2

BL2p

NL3p, fail test4 list\_nesting

Test 5: BL1 / p > NL1 Warn

* BL1

1. NL1 Warn, warn test5 list\_warn

Test 6: BL1 / p > BL2 OK

* BL1

BL1p

* BL2 pass

Test 7: BL3 / p > NL2 conditional pass

* BL1

BL1p

Extract4

* 1. NL2

Nl2p

Extract1

* BL3

BL3p

* 1. NL2 pass

Test 8: BL3 / p > NL2 conditional Not OK

* BL1

BL1p

Extract4

* BL2

Extract1

* BL3

BL3p

* 1. NL2, fail test 8 list\_change

Test 9: BL1 / p > BL3 Not OK

* BL1
* BL3 fail test 9 list\_nesting

Test 10: BL3 / p > NL3 Not OK

* BL1
  1. AL2
* BL3
  + 1. NL3 fail test 10 list\_change

Test 11: Non-list > BL2 Not OK

ExHead

Extract2

* BL2 fail test 11 list\_nesting

BL2p

Test 12: Non-list > Extract > AL1 OK

ExHead

Extract2

1. AL1

AL1p

Test 13: BL2p > NL3 OK

1. AL1

* BL2

BL2p

* + 1. NL3

Test 14: BL1 > NL2p Not OK

* BL1

NL2p fail test 13 list\_nesting

Test 15: BL2 / p > BL1 conditional OK

* BL1
* BL2

Ext3

BL2p

* BL1

Test 16: Nonlist > AL1p Not OK

AL1p fail list\_nesting

Test 17: Nonlist > AL1p Not OK

Extract-1

AL1p fail list\_nesting