

U051SS2EU : it's a different class, but has the same behaviors

U051SS2EU : another alternative is to use composition (a deftype with a Foo, which re-calls every method on that Foo)

U0EHU1800 : Thanks for the alternatives. I'll ponder those for a bit to see if I can make them work. Cheers!

U0CKDHF4L : what's wrong with this spec expression that it doesn't return an explanation ? ```(s/explain (s/cat :this (s/* (s/coll-of :q/o)) :that :q/test-map) '((7 3 1) (9 7 3) {:q/b [2 4 6] :q/c ["abc" "xyz"] :q/d [:q :w] :q/a [1 3 5]}))IllegalArgumentException Argument must be an integer: [:q/b [2 4 6]] clojure.core/even? (core.clj:1383)```

U0CKDHF4L : ...each part of the s/cat works: ```(s/explain (s/cat :this (s/* (s/coll-of :q/o))) '((7 3 1) (9 7 3)))Success!```

U0CKDHF4L : ```(s/explain (s/cat :that :q/test-map) '({:q/b [2 4 6] :q/c ["abc" "xyz"] :q/d [:q :w] :q/a [1 3 5]}))Success!```

U0CKDHF4L : (where ```(s/def :q/test-map (s/keys :req [:q/b :q/a] :opt [:q/c :q/d]))``` and the various keys :q/a etc are simple ```odd? even? string? keyword?```

U0NCTKEV8 : even? isn't total

U0NCTKEV8 : it throws an exception if something isn't a number

U0NCTKEV8 : so you need something like (s/and number? even?)

U0CKDHF4L : sorry I should have said :q/a and q:b are ```(s/+ odd?)``` and ```(s/+ even?)```

U0NCTKEV8 : what is :q/o ?

U0CKDHF4L : ```(s/def :q/o odd?)```

U0NCTKEV8 : so the way s/* can tell it is done matching is something fails to match

U0NCTKEV8 : yeah, odd? isn't total

U6902JG68 : Hey guys can someone help with this problem, I am new to clojure, may be I am missing something:<<https://stackoverflow.com/questions/45107883/clojure-reducers-foldcat-not-working>>

U0NCTKEV8 : (s/and number? odd?)

U0CKDHF4L : what do you mean by "total" ?

U04V70XH6 : I suspect the problem here is that `s/+` is a regex matcher, not a collection matcher...?

U051SS2EU : <@U0CKDHF4L> <<http://mathworld.wolfram.com/TotalFunction.html>>

U04V70XH6 : Change `:q/b` to `(s/coll-of even?)`

U04V70XH6 : (you can specify a minimum length of 1 to require at least one element)