U08JL5H89 : <@U0702F2CE> That makes sense for one direction. But then it doesn't make sense why when its done at phase 1 the vector becomes mutable.
U0702F2CE : <@U08JL5H89> I agree, I don't understand what's going on there
U0702F2CE : but that should be investigated without using 3 layers of quoting
U08JL5H89 : <@U0702F2CE> Ha, ya, I do agree with that.
U0702F2CE : and is unrelated to the typing issue
U08JL5H89 : I also agree with that.
U08JL5H89 : (I forgot that this originated from a typed/racket thing. <@U3QF0EM0E> and I were discussing it (outside of slack) without the context of TR.)
U08JL5H89 : <@U3NJS8H7C> (Or anyone else familiar enough with the FFI to answer), I have the following program that uses ffi/unsafe/alloc: <https://gist.github.com/LeifAndersen/353e0da21d9213dcd6a79cbd9b8c94ee>
U08JL5H89 : In a loop, it allocates a frame, relying on the GC to clean up.
U08JL5H89 : However, it leaks memory very quickly.
U3QF0EM0E : yes <@U0702F2CE> thanks for pointing out `datum->syntax`. Here's another phase 1 example (still returns the weird result)```
#lang racket/base
(require (for-syntax racket/base))

(define-syntax (foo stx)
  (with-syntax ((x (datum->syntax stx (vector-immutable 1))))
   (syntax (syntax x))))

(immutable? (syntax-e (foo)))
```

U08JL5H89 : BUT, if I remove line 15 (the deallocator wrapper), it stops leaking memory.
U08JL5H89 : Does anyone have any suggestions?
U0702F2CE : <@U3QF0EM0E> here's the bug:
U08JL5H89 : (For the record, if I remove both alloc and dealloc, and just call the `av-frame-free` function by hand everything works fine.)
U0702F2CE : ```&gt; (immutable? (syntax-e (quote-syntax #(1))))
#f
```

U0702F2CE : note that `syntax->datum` does the right thing
U3NJS8H7C : leif: The short answer is that the finalization system used by `allocator` is not good enough (and wouldn't be good enough in most runtimes). Finalizers via `allocator` are run in a separate thread, so the program requires a combination of a GC and a context switch and another GC before some relevant memory can be reclaimed. Some solutions might involve limiting the allocator via `(sleep)` or forcing an occasional GC via `(collect-garbage)`.
U3NJS8H7C : I now see the rest of your original message, and I'm puzzled offhand that removing `#:wrap (deallocator)` helps, so I'll investigate a little more.
U08JL5H89 : <@U0702F2CE> and <@U3QF0EM0E> FWIW, I ran this test on Racket7, and got:```
leif@FATT ~/src/racket7/racket/bin (master) $ ./racket test.rkt
----
FAILURE
name: check-false
location: test.rkt:4:0
params: '(#t)
message: "(syntax-e (syntax (vector))) made immutable vector"

FAILURE
name: check-false
location: test.rkt:19:0
params: '(#t)
message: "(syntax-e (syntax (vector))) made immutable vector"

```

U08JL5H89 : So I'm pretty sure there is a bug here.
```

U3QF0EM0E : ^ this test = <http://pasterack.org/pastes/86496>
U3QF0EM0E : (point is, now the failures are consistent)
U3NJS8H7C : As far as I can tell, adding `#:wrap (deallocator)` slows down `av-frame-free` enough that the finalization thread moves more slowly than the allocation thread, so my original answer is still what I think is happening. Removing `#:wrap (deallocator)` might be a reasonable solution if `av-frame-free` is not to be called directly.
U08JL5H89 : Ah, okay, thanks makes a lot of sense. Thank you.