

[New issue](#)[Jump to bottom](#)

# Stack overflow due to recursion in src/dfa/dead\_rules.cc #394

✓ Closed Me19m4 opened this issue on Jan 20 · 2 comments

Me19m4 commented on Jan 20

Operating System Version: ubuntu 20.04

re2c version: 2.2

error function: re2c::backprop

==9992==ERROR: AddressSanitizer: stack-overflow on address 0x7ffdf3f83ff8 (pc 0x00000066f8e0 bp 0x0000000135534 sp 0x7ffdf3f84000 T0)

#0 0x66f8e0 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

#1 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

#2 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

#3 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

Omit.....

#245 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

#246 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

#247 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

#248 0x66f8e4 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)  
re2c/src/dfa/dead\_rules.cc:149:9

AddressSanitizer: stack-overflow re2c/src/dfa/dead\_rules.cc:149:9 in re2c::backprop(re2c::rdfa\_t const&, bool\*, unsigned long, unsigned long)

Test example link:

<https://drive.google.com/file/d/1bLXgifNQhcTQl6937lJhapAa3hgwEugT/view?usp=sharing>

Run the following command to repeat the error:

```
$ ./re2c example
```

skvadrik commented on Jan 20

Owner



Thanks for the bug report. Did you find these examples with some kind of fuzzer?

There are two different places where re2c should enforce reasonable size limits:

- NFA size and depth
- DFA size and depth

For regular expressions it is not necessary, because counted repetition is only unrolled when NFA is constructed (so RE can't get much larger than their text representation in the source file). For NFA and DFA the limits should be enforced separately, because there may be very large NFA that result in very small DFA (e.g. for something like `((("){0,100}){0,100}){0,100}` NFA will have about  $100^3$  states, but DFA will have just one state). And the other way around (a small NFA triggering pathological exponential DFA size).

In the second test case re2c should check that the lower repetition bound is less or equal to the upper bound.

  skvadrik changed the title ~~Stack overflow due to recursion in re2c/SRC/dfa/dead\_rules.cc~~ Stack overflow due to recursion in src/dfa/dead\_rules.cc on Jan 21

 skvadrik added a commit that referenced this issue on Jan 21

 Limit maximum allowed NFA and DFA size. ...  a3473fd


 skvadrik added a commit that referenced this issue on Jan 21

 Emit an error when repetition lower bound exceeds upper bound. ...  039c189

skvadrik commented on Jan 21

Owner

Fixed in commits [a3473fd](#) and [039c189](#) .

 skvadrik closed this as completed on Jan 23

No one assigned

---

#### Labels

None yet

---

#### Projects

None yet

---

#### Milestone

No milestone

---

#### Development

No branches or pull requests

---

2 participants

