



Products

E0597 trying to loop over an vector

Log in

Sign up

Ask Question

Asked 10 months ago

Active 10 months ago

Viewed 62 times



1



I'm teaching myself Rust by doing some programming exercises. Here I am have a `Vec<String>` I read out of a file, and I am trying to put the letters in each line in a `HashSet`, then taking the intersection of all the lines.

When I do this I get an error saying my loop variable does not live long enough.

I have an intuition what is wrong -- I have a variable `l` whose lifespan is one iteration of the loop, and another variable `candidates` whose lifespan is the entire loop, and the second variable is borrowing from the first. But how do I fix this?

```
let mut candidates = HashSet:::<&u8>::new();
let mut sum = 0;
for l in lines { // lines is Vec<String>
    if candidates.len() == 0 {
        candidates = HashSet:::<&u8>::from_iter(l.as_bytes());
        println!("candidates = {:?}", candidates);
    } else if l.len() == 0 { // error message "borrow later used here"
        // end of group
        sum = sum + candidates.len();
        candidates = HashSet:::<&u8>::new();
    } else {
        let h2 = HashSet:::<&u8>::from_iter(l.as_bytes()); // error message "borrowed value does not live long enough"
        candidates = candidates.intersection(&h2).copied().collect();
        println!("candidates = {:?}", candidates);
    }
    println!("{}", l);
}
```

I tried making a copy of `l`

```
let mut l2:String = "".to_string();
for l in lines {
    if candidates.len() == 0 {
        l2 = l.to_string();
        candidates = HashSet:::<&u8>::from_iter(l2.as_bytes());
    }
}
```

but I get a different error saying `error[E0506]: cannot assign to l2 because it is borrowed`.

[loops](#) [rust](#) [set](#) [lifetime](#) [borrow-checker](#)

Share

Improve this question

Follow

edited Feb 11 at 0:15



pretzelhammer

11.7k • 14 • 39 • 88

asked Feb 10 at 22:27



benhsu

4,946 • 8 • 35 • 46

1

can you just borrow as in `for l in &lines { ?`

-- oliver

Feb 10 at 23:07

Add a comment

2 Answers

[Active](#) [Oldest](#) [Votes](#)



3



`candidates` can contain references into `l` from the previous loop iteration, but `l` is dropped at the end of each loop iteration, hence the compile errors. You can avoid all this trouble by using `into_bytes` instead of `as_bytes` so you get owned values instead of references. Here's a cleaned up refactored version:

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Customize settings

```

use std::collections::HashSet;

fn example(lines: Vec<String>) {
    let mut candidates: HashSet<u8> = HashSet::new();
    let mut sum = 0;

    for line in lines {
        dbg!(&line);
        let h2: HashSet<u8> = line.into_bytes().into_iter().collect();
        if candidates.is_empty() {
            candidates = h2;
        } else if h2.is_empty() {
            sum += candidates.len();
            candidates = HashSet::new();
        } else {
            candidates = candidates.intersection(&h2).copied().collect();
        }
        dbg!(&candidates);
    }
}

```

[playground](#)

[Share](#)

[Improve this answer](#)

[Follow](#)

answered Feb 10 at 23:17



[pretzelhammer](#)

11.7k • 14 • 39 • 88

2

Another advantage of this solution is that `u8` is smaller and faster than `&u8`.

– [Jnb](#)

Feb 11 at 7:31

[Add a comment](#)



2



`.as_bytes` converts your `String` to a byte slice. Slices are references to a contiguous sequence of elements. The compiler points out that at the end of your for loop, the reference is invalid.

One solution would be to iter over `String` references with `&lines`:

```

let mut candidates = HashSet::<u8>::new();
let mut sum = 0;
for l in &lines { // <= HERE iter over String references which lives outside the for loop
    if candidates.len() == 0 {
        candidates = HashSet::<u8>::from_iter(l.as_bytes());
        println!("candidates = {:?}", candidates);
    } else if l.len() == 0 {
        // end of group
        sum = sum + candidates.len();
        candidates = HashSet::<u8>::new();
    } else {
        let h2 = HashSet::<u8>::from_iter(l.as_bytes());
        candidates = candidates.intersection(&h2).copied().collect();
        println!("candidates = {:?}", candidates);
    }
    println!("{}", l);
}

```

A second solution would be to copy your `String`.

```

let mut candidates = HashSet::<u8>::new();
let mut sum = 0;
for l in lines { // lines is Vec<String>
    if candidates.len() == 0 {
        candidates = HashSet::<u8>::from_iter(l.clone().into_bytes());
        println!("candidates = {:?}", candidates);
    } else if l.len() == 0 {
        // end of group
        sum = sum + candidates.len();
        candidates = HashSet::<u8>::new();
    } else {
        let h2 = HashSet::<u8>::from_iter(l.clone().into_bytes());
        candidates = candidates.intersection(&h2).copied().collect();
        println!("candidates = {:?}", candidates);
    }
    println!("{}", l);
}

```

[Share](#)

[Improve this answer](#)

[Follow](#)

answered Feb 10 at 23:11



[Quentin Baradat](#)

91 • 4

[Add a comment](#)



Your Answer

Post Your Answer



By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [loops](#) [rust](#) [set](#) [lifetime](#) [borrow-checker](#) or ask your own question.

The Overflow Blog

-  Sequencing your DNA with a USB dongle and open source code
-  Don't push that button: Exploring the software that flies SpaceX rockets and...

Featured on Meta

-  Providing a JavaScript API for userscripts
-  Congratulations to the 59 sites that just left Beta

Related

[C# loop - break vs. continue](#)

[How do I loop through or enumerate a JavaScript object?](#)

[How to loop through a plain JavaScript object with the objects as members](#)

[A 'for' loop to iterate over an enum in Java](#)

[3621](#)

[Loop through an array in JavaScript](#)






[For-each over an array in JavaScript](#)

["borrowed value does not live long enough" when using the builder pattern](#)

[Problems with Tuple's lifetime in rust.](#)

[why is rust 'pub fn func\(&'a mut self\)' considered "mutably borrowed" after run?](#)

Hot Network Questions

-  [Stabilizers in a permutation group](#)
-  ['Cloth shop' and 'Clothes shop'](#)
-  [Why are nerves blocked even though potassium channels are not blocked?](#)
-  [How can I let my opponent know that I'm touching a piece to adjust it?](#)
-  [What do I do when my boss is sabotaging interviews?](#)

[more hot questions](#)

 [Question feed](#)

STACK OVERFLOW

[Questions](#)
[Jobs](#)
[Developer Jobs Directory](#)
[Salary Calculator](#)
[Help](#)
[Mobile](#)

PRODUCTS

[Teams](#)
[Talent](#)
[Advertising](#)
[Enterprise](#)

COMPANY

[About](#)
[Press](#)
[Work Here](#)
[Legal](#)
[Privacy Policy](#)
[Terms of Service](#)
[Contact Us](#)
[Cookie Settings](#)
[Cookie Policy](#)

STACK EXCHANGE NETWORK

[Technology](#)
[Culture & recreation](#)
[Life & arts](#)
[Science](#)
[Professional](#)
[Business](#)
[API](#)
[Data](#)

[Blog](#)
[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)

site design / logo © 2021 Stack Exchange Inc; user contributions licensed under [cc by-sa](#). rev 2021.12.22.41046