Lifetime parameters in supertrait bounds

Ask Question

Asked 1 year, 8 months ago
Active 1 year, 8 months ago
Viewed 115 times

2

I'm trying to define a trait for an object that is convertible to and from a slice of bytes. I essentially want to say

```
trait Foo: AsRef<[u8]> + TryFrom<&[u8]> {}
```

Unfortunately, this refuses to compile unless I put a lifetime parameter on the reference, like so:

```
trait Foo<'a>: AsRef<[u8]> + TryFrom<&'a [u8]> {}
```

This doesn't make much sense to me, because the lifetime `'a` is related to the eventual `try_from()` call and shouldn't have to be part of the object's type. (The implementation of `try_from()` copies the relevant bytes, so the lifetime of its parameter really isn't relevant.)

This seems like a more general issue than just slices, though; how do you specify lifetime parameters like this for supertrait bounds? (Apparently `'_` doesn't work.) And is there a better/more idiomatic way to express this, or do I have to resort to some sort of hand-rolled custom nonsense like

```
pub trait TryFromRef<T> { type Error; fn try_from(value: &T) -> Result<Self, Self::Error>; }
```

?

rust traits lifetime

Share
Improve this question
Follow

asked Apr 10 '20 at 1:39

Reid Rankin
997 ● 6 ● 25

There must be a duplicate for this one, I'll look for it. The answer is to use a *higher-ranked trait bound*: `trait Foo: AsRef<[u8]> + for<'a> TryFrom<&'a [u8]> {}`
– trent formerly cl
Apr 10 '20 at 1:48

@trentcl Genius. I remember reading about those in the Book, but didn't connect it with this issue. Stick it in an answer and I'll accept
– Reid Rankin
Apr 10 '20 at 1:50

Yeah, the other questions I was thinking of aren't exactly what you're looking for IMO. I'll write an answer.
– trent formerly cl
Apr 10 '20 at 1:56

Add a comment

1 Answer

Active Oldest Votes

1

A trait bound with a lifetime parameter that holds for all lifetimes, rather than for some particular lifetime, can be specified with a so-called *higher-ranked trait bound*, or HRTB. In your case this might look like

```
trait Foo: AsRef<[u8]> + for<'a> TryFrom<&'a [u8]> {}
```

Anything implementing `Foo` must satisfy `TryFrom<&'a u8>` for any and all choices of `'a`, so there's no need for a lifetime on `Foo` itself.

See also

- How do I write the lifetimes for references in a type constraint when one of them is a local reference?
- How does for<> syntax differ from a regular lifetime bound?

Share
Improve this answer

Follow

answered Apr 10 '20 at 2:05

trent formerly cl

20.1k ● 7 ● 42 ● 72

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 rust traits lifetime 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

Linked

How does for<> syntax differ from a regular lifetime bound?

How do I write the lifetimes for references in a type constraint when one of them is a local reference?

Related

What is the lifetime of a static variable in a C++ function?

14
Closures and Higher-Ranked-Trait-Bounds lifetime issue

Issues constraining implementation lifetimes on type without lifetime parameter

Define a trait with a function that returns an associated type with the same lifetime as one parameter

Specify `Fn` trait bound on struct definition without fixing one of the `Fn` parameters

How to put explicit lifetime bound on Self for associated constant?

Lifetime sub-typing and impl-trait

In Rust, when boxing a value passed as a generic argument, why is a "static` lifetime bound required?

Hot Network Questions

- Will these simple 2 convex lens arrangement telescope see the moon clearly?
- How do I get the http endpoint to work for cardano-wallet?
- Is there a deterministic guide to landing?
- What does this entry on the Rocinante's pilot quick-menu mean?
- Is it possible to convert a taproot address into a native segwit address?

more hot questions

Question feed