

# Self-referential structs

AND HOW  
TO CREATE THEM

Wojciech Polak  
December 2019



# The story

# The story

- What I wanted to achieve

# The story

- What I wanted to achieve
- What problems I stumbled upon

# The story

- What I wanted to achieve
- What problems I stumbled upon
- My idea with **unsafe** code

# The story

- What I wanted to achieve
- What problems I stumbled upon
- My idea with **unsafe** code
- How /r/rust helped me to find solution

# Markdown parser

# Markdown parser

- I wanted to parse some markdown



# Markdown parser

- I wanted to parse some markdown
- But I didn't want to render html instantly

# Markdown parser

- I wanted to parse some markdown
- But I didn't want to render html instantly
- I liked crate: [raphlinus/pulldown-cmark](https://crates.io/crates/raphlinus/pulldown-cmark)

# Markdown parser

- I wanted to parse some markdown
- But I didn't want to render html instantly
- I liked crate: [raphlinus/pulldown-cmark](https://crates.io/crates/raphlinus/pulldown-cmark)
- It's fast, powerful and returns iterator over *Event*`<'a>`

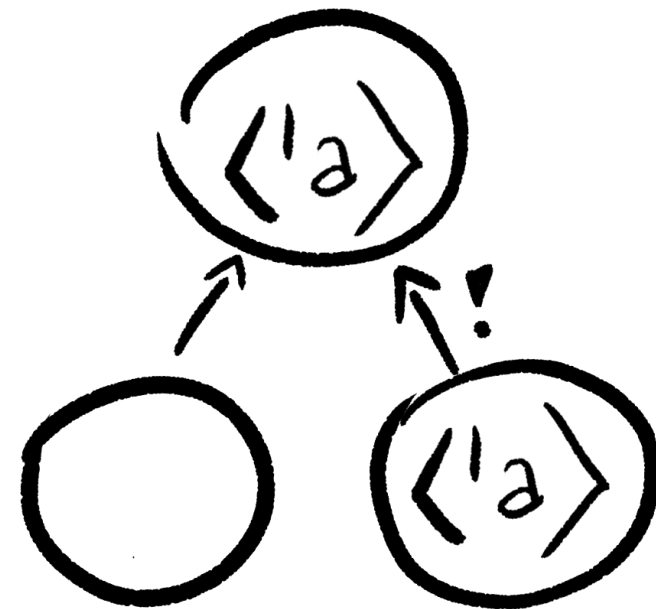
# The problem

# The problem

- Lifetimes are contagious!  
(unless static)

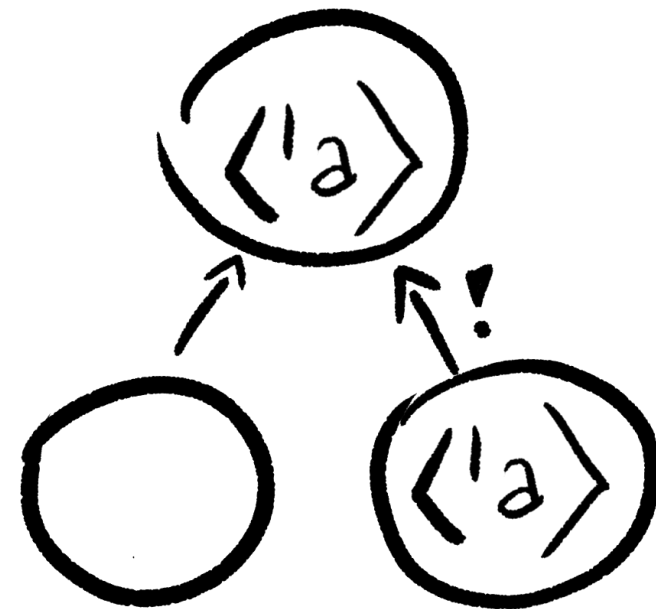
# The problem

- Lifetimes are contagious!  
(unless static)



# The problem

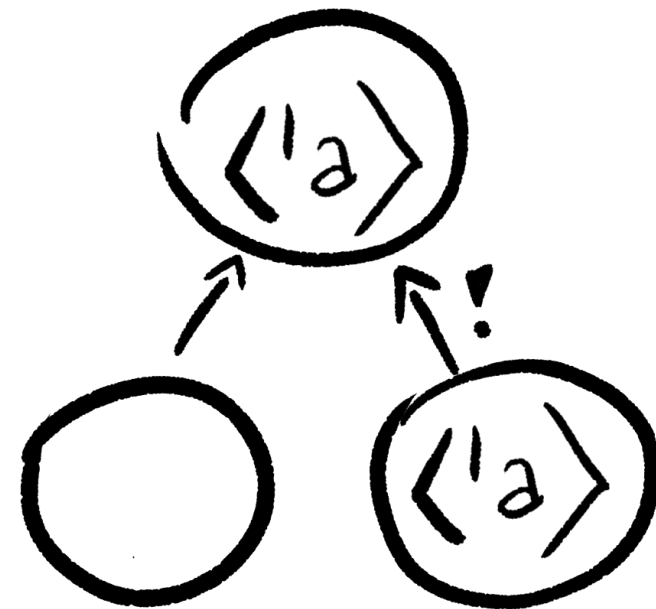
- Lifetimes are contagious!  
(unless static)



- And I wanted to use Salsa crate: [salsa-rs/salsa](https://crates.io/crates/salsa)

# The problem

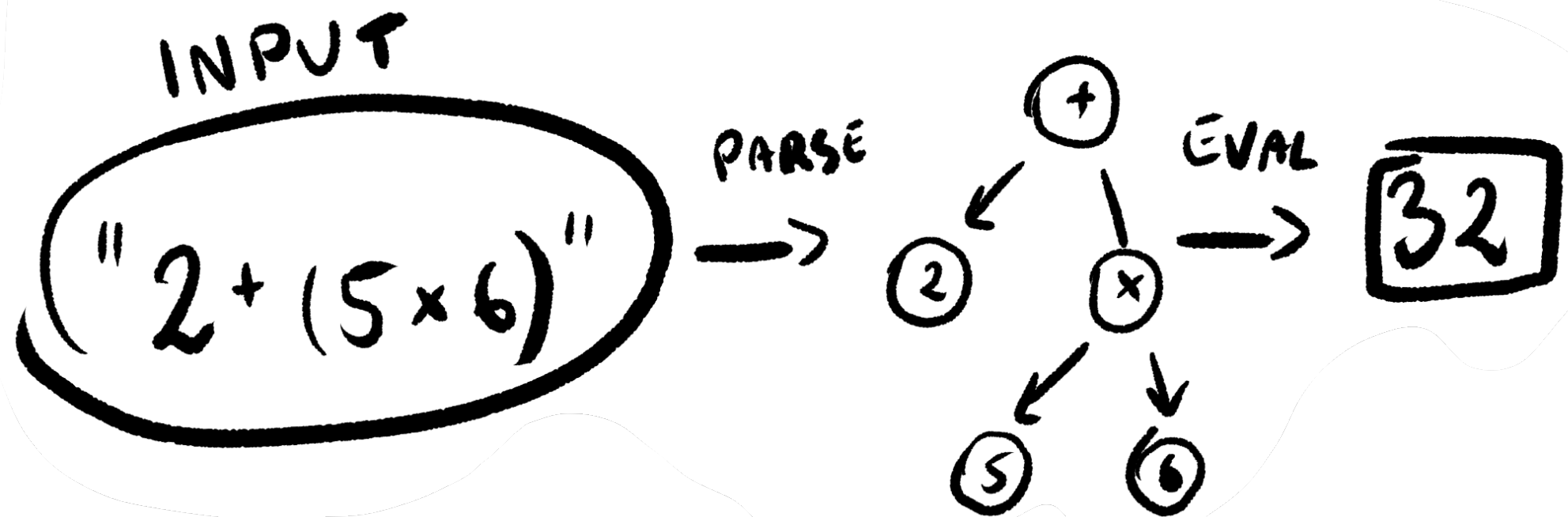
- Lifetimes are contagious!  
(unless static)



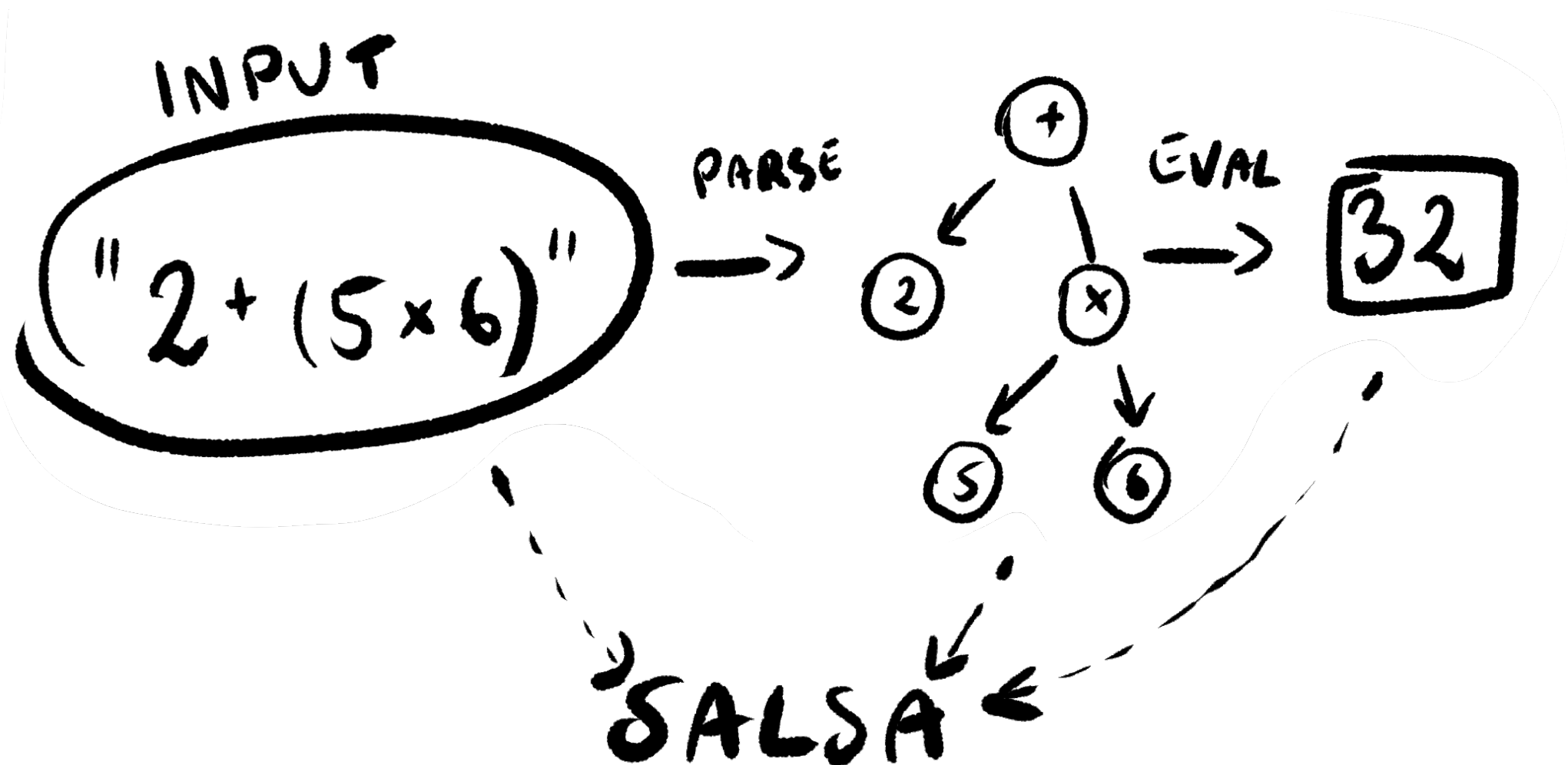
- And I wanted to use Salsa crate: [salsa-rs/salsa](https://crates.io/crates/salsa)
- And it doesn't support lifetimes right now :(



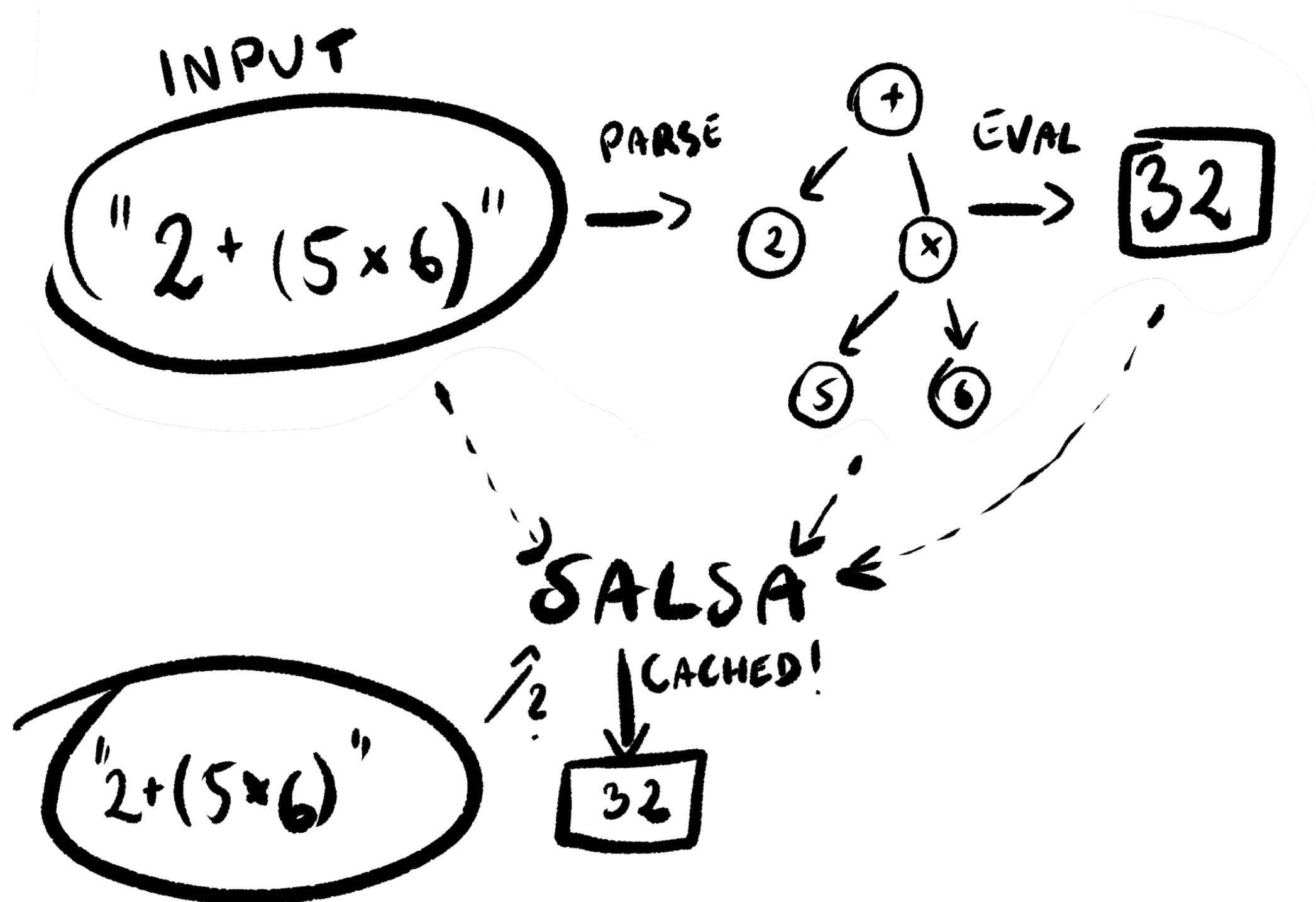
# But what is Salsa (briefly)



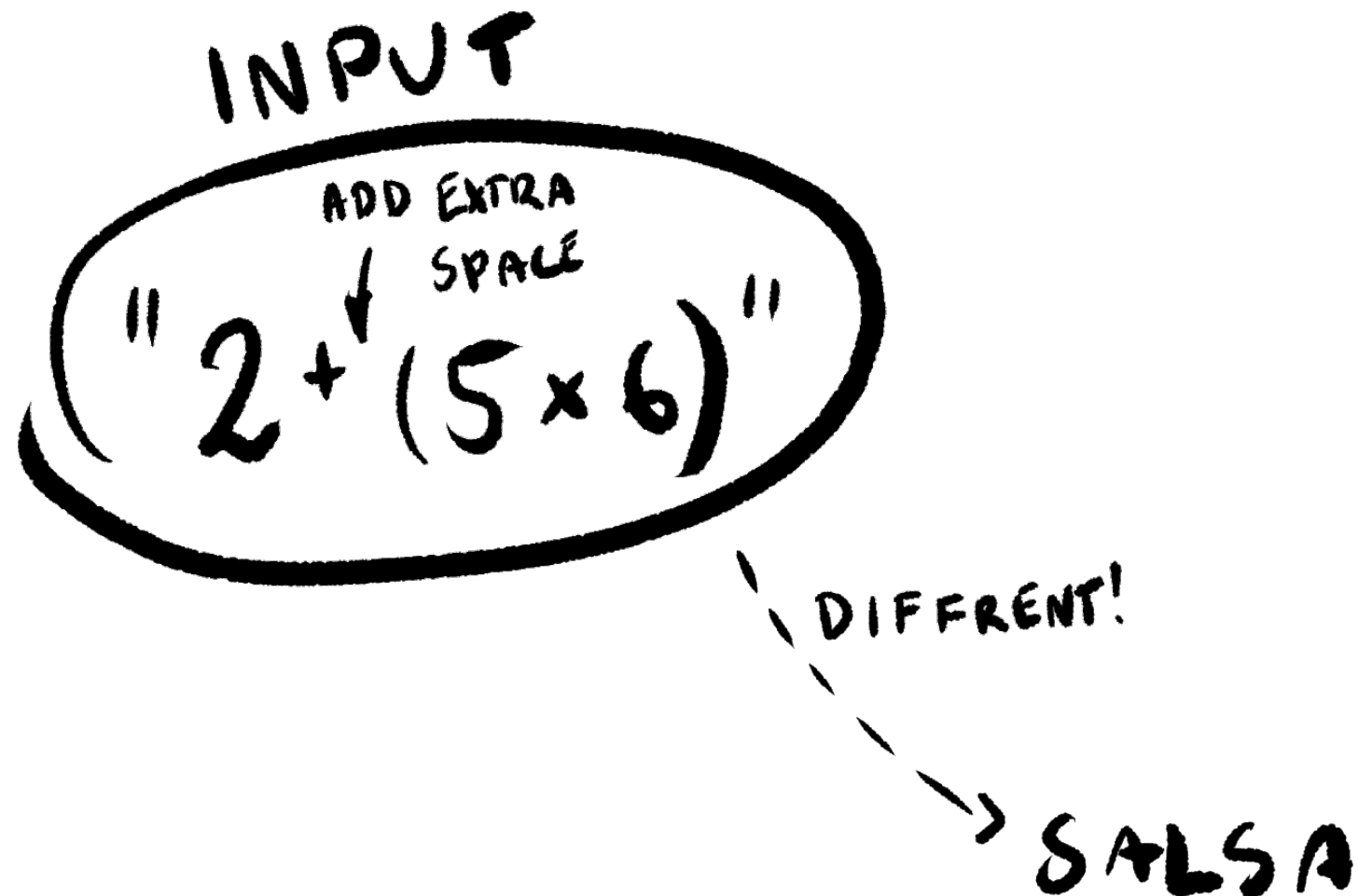
# But what is Salsa (briefly)



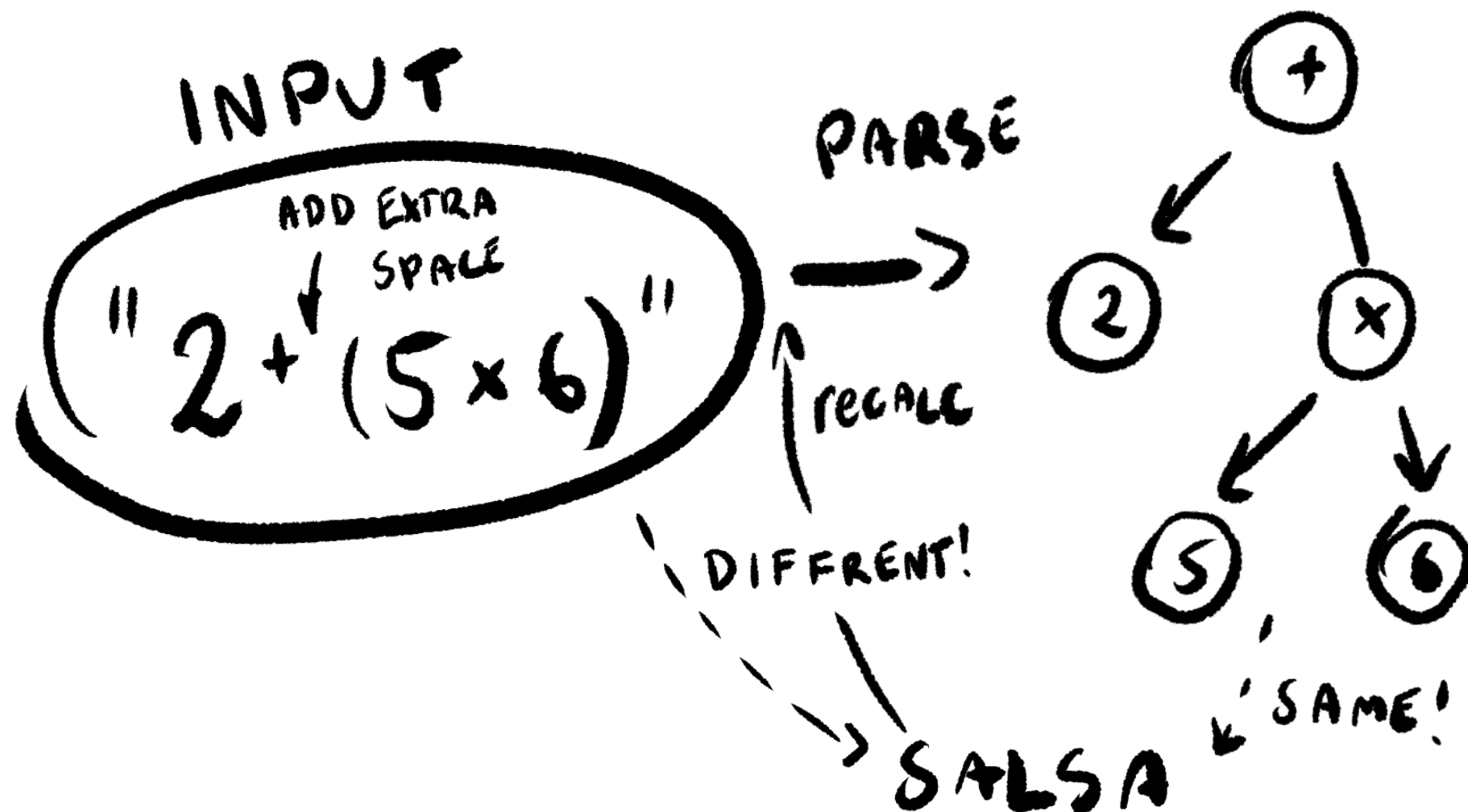
# But what is Salsa (briefly)



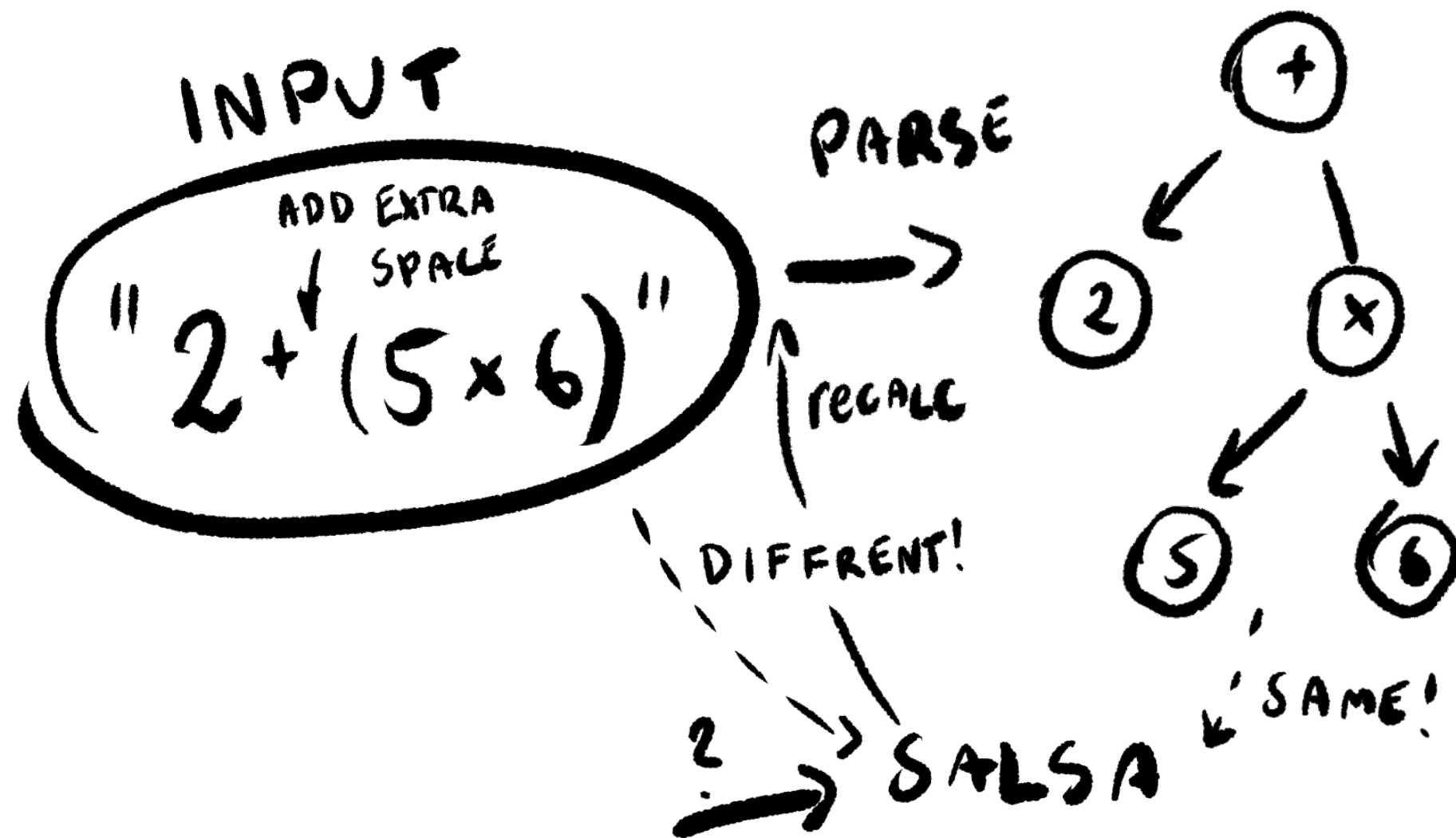
# But what is Salsa (briefly)



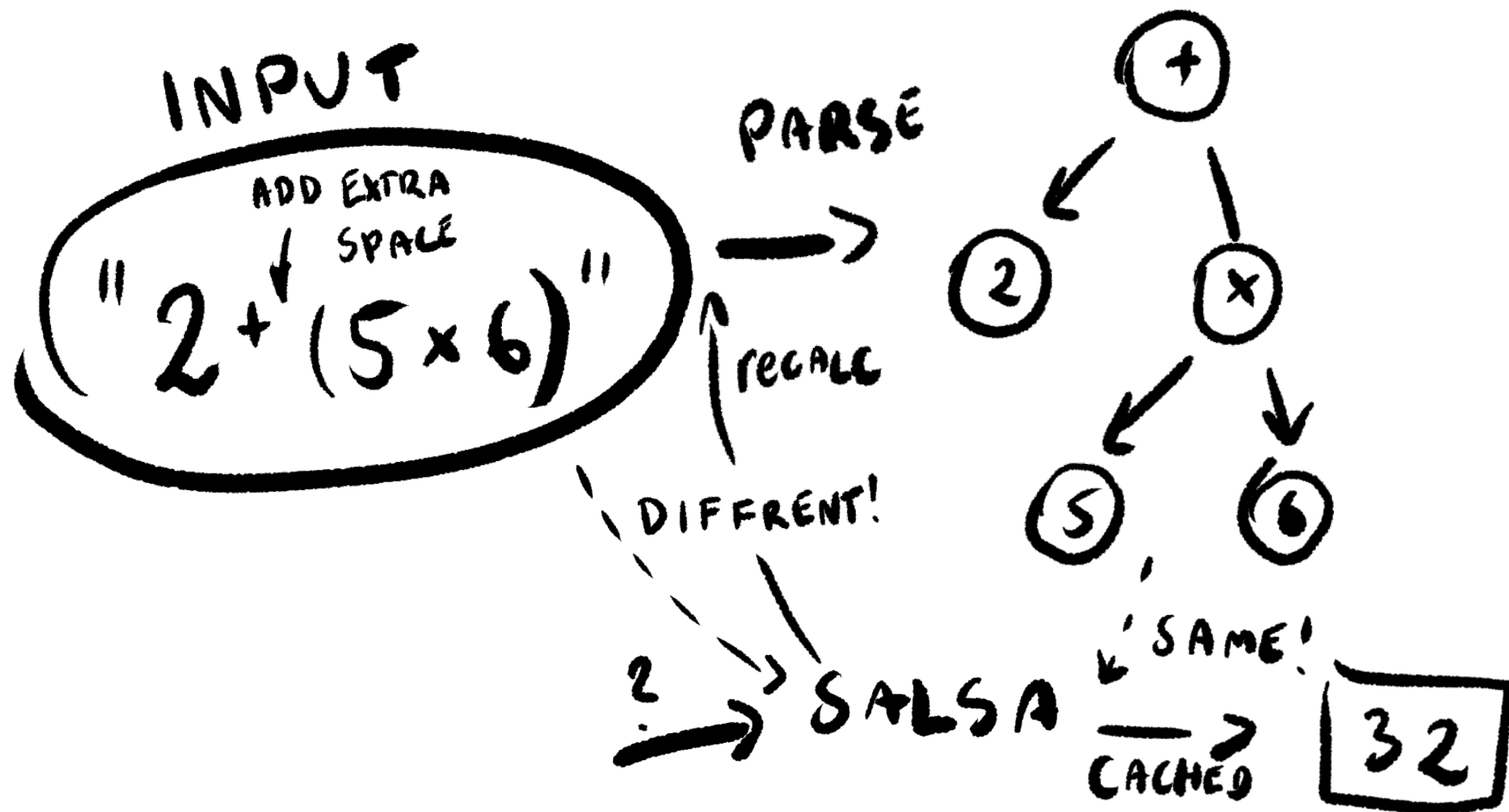
# But what is Salsa (briefly)



# But what is Salsa (briefly)



# But what is Salsa (briefly)



**“And it doesn’t support lifetimes right now :(”**

*–Wojciech Polak*



# Example of problem

[https://play.rust-lang.org/?](https://play.rust-lang.org/?version=stable&mode=debug&edition=2018&gist=9483d6200925aab03694046391105d90)

[version=stable&mode=debug&edition=2018&gist=9483d6200925aab03694046391105d90](https://play.rust-lang.org/?version=stable&mode=debug&edition=2018&gist=9483d6200925aab03694046391105d90)

# My solution

# My solution

- Let's hide lifetime by using self-referential struct

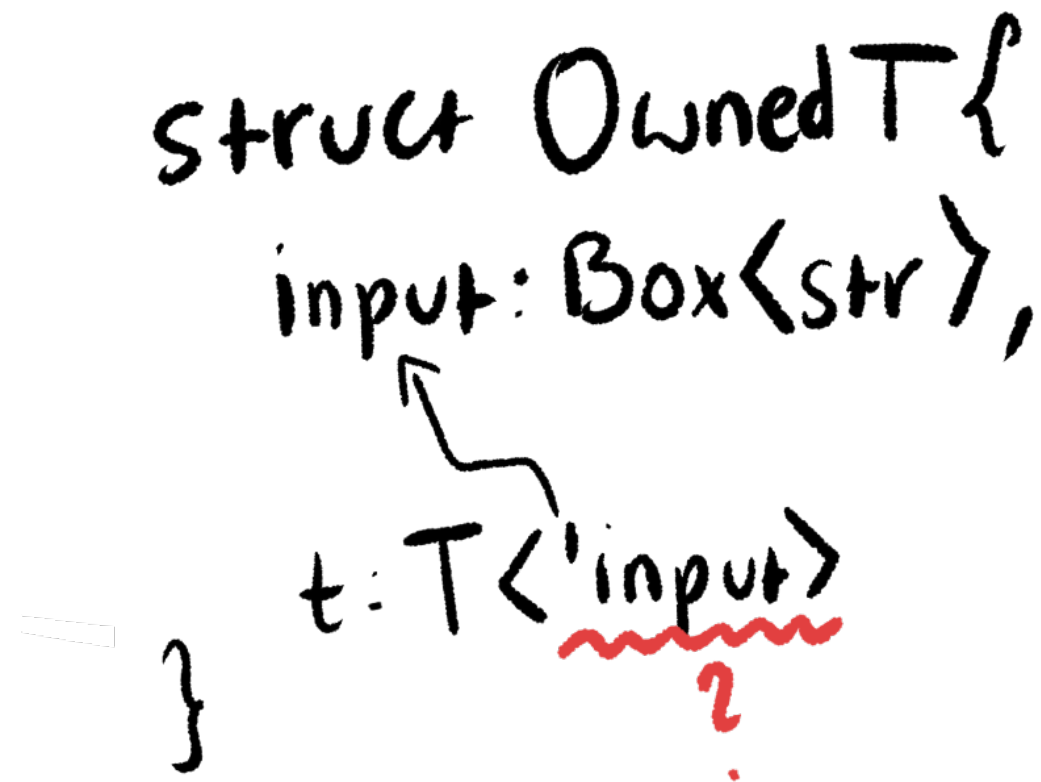
# My solution

- Let's hide lifetime by using self-referential struct
- With a little dose of **unsafe**

# My solution

- Let's hide lifetime by using self-referential struct
- With a little dose of **unsafe**

```
struct OwnedT {  
    input: Box<str>,  
    t: T<input>  
}
```




# My solution

- Let's hide lifetime by using self-referential struct
- With a little dose of **unsafe**

```
struct OwnedT {  
    input: Box<str>,  
    t: T<'input>?  
}
```

static + unsafe



# My solution

[https://play.rust-lang.org/?  
version=stable&mode=debug&edition=2018&gist=2ce2b0efb  
d7ecad252b11b8c6846d86b](https://play.rust-lang.org/?version=stable&mode=debug&edition=2018&gist=2ce2b0efbd7ecad252b11b8c6846d86b)



# Is it sound?

Or did I made a mistake?



↑ jDomantas 2 points · 14 days ago

↓ OwnedT::same\_source is unsound, as it exposes a T<'static> that borrows a non-static string:

```
let evil: T<'static> = {  
    let cell = std::cell::Cell::new(None);  
    let x = OwnedT::create("oh noes", |f| f.into());  
    x.same_source(|s| {  
        cell.set(Some(s.child()));  
        s.child()  
    });  
    cell.into_inner().unwrap()  
};  
dbg!(evil);
```

# Reddit will tell you

If you ask nicely ;)

# Found problems

# Found problems

- same\_source is unsound

# Found problems

- `same_source` is unsound
- There is undefined behaviour during drop

# Found problems

- `same_source` is unsound
- There is undefined behaviour during drop
- Potential modification of `Arc<>`

# Found problems

- `same_source` is unsound
- There is undefined behaviour during drop
- Potential modification of `Arc<>`
- ... not a problem but nice suggested resource:  
<https://doc.rust-lang.org/nomicon/meet-safe-and-unsafe.html>

# Working example

[https://play.rust-lang.org/?](https://play.rust-lang.org/?version=stable&mode=debug&edition=2018&gist=cca379bd)

[version=stable&mode=debug&edition=2018&gist=cca379bd](https://play.rust-lang.org/?version=stable&mode=debug&edition=2018&gist=cca379bd)  
[c24092ea6adc030a0c163797](https://play.rust-lang.org/?version=stable&mode=debug&edition=2018&gist=cca379bd)

# Light of hope



# Light of hope

- <https://nikomatsakis.github.io/rust-belt-rust-2019/#1>

# Light of hope

- <https://nikomatsakis.github.io/rust-belt-rust-2019/#1>
- **Salsa creators have plans for supporting lifetimes (but in rather far future)**

# Summary

# Summary

- Don't be afraid of unsafe...

# Summary

- Don't be afraid of unsafe...
- ...But make sure you need it,

# Summary

- Don't be afraid of unsafe...
- ...But make sure you need it,
- And you know what are you doing.

# Summary

- Don't be afraid of unsafe...
- ...But make sure you need it,
- And you know what are you doing.
- And ask! Ask questions. This community is amazing ;)

# Fin

