

# Dependent Types in the Idris Programming Language

## Part 3: Streams and Interaction

Edwin Brady ([ecb10@st-andrews.ac.uk](mailto:ecb10@st-andrews.ac.uk))  
University of St Andrews, Scotland, UK  
[@edwinbrady](https://twitter.com/edwinbrady)

OPLSS, July 3rd 2017

- What limitations does totality impose for practical programming?

- What limitations does totality impose for practical programming?
- Is it possible to write partial functions and prove totality later?

- What limitations does totality impose for practical programming?
- Is it possible to write partial functions and prove totality later?
- How is `impossible` defined? Can I use it in a `case` expression?

- What limitations does totality impose for practical programming?
- Is it possible to write partial functions and prove totality later?
- How is `impossible` defined? Can I use it in a `case` expression?
- How does Idris deal with Universes? (i.e. what is the type of `Type`?)

- Can you say more about how erasure works?

- Can you say more about how erasure works?
- Where can I get a t-shirt that says “It type checks! Ship it!”

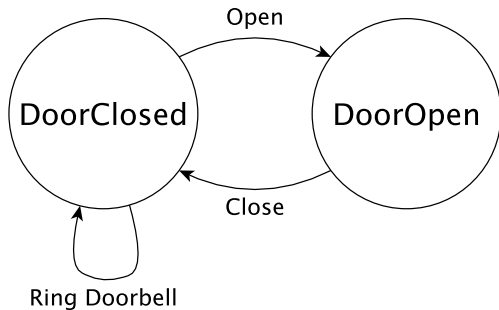
- Can you say more about how erasure works?
- Where can I get a t-shirt that says “It type checks! Ship it!”
- Are you Scottish, because you don't sound Scottish?



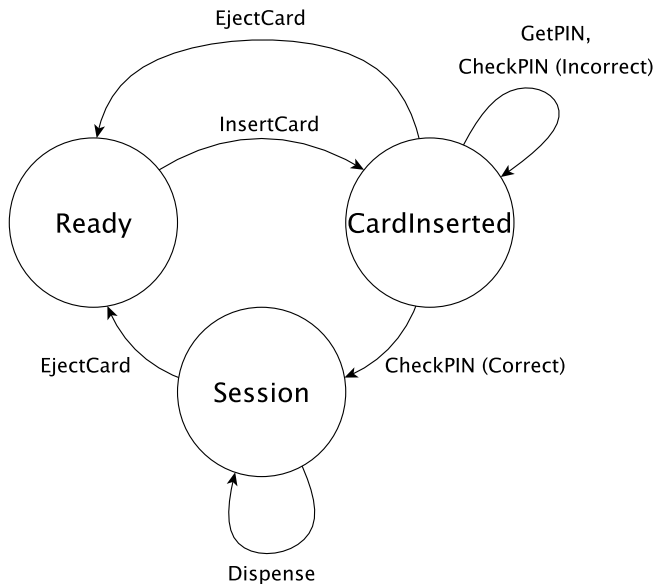


Demonstration: Streams and Interaction

# Example State Machine: Opening a Door



# States in a Bank's ATM





Demonstration: State Machines