Formal verification for the masses

Testing vs Proving

```
2 + 2                    |        x + x
= { addition rules }     |    = { basic algebra }
4                        |        2 * x
```

Rules for transforming programs

```
  if ¬true then A else B
= { applying ¬ }
  if false then A else B
= { semantics of the if statement }
  B
```

Smart contracts

- It's important to sleep at night if you're responsible for a smart contract holding valuable assets

- Formal verification can and has been applied successfully to make more secure smart contracts

Formal Land

- Ethereum, Tezos, Sui, Aleph Zero

Why it's not used more?

- Dependent types

- Niche platforms and languages

- PhD level knowledge

```
1 +1-length : ∀ {ℓ}{A : Set ℓ}(h : A)(t : 𝕃 A) → length (h ∷ t) ≡ length t + 1
2 +1-length h [] = refl
3 +1-length h t = refl
```
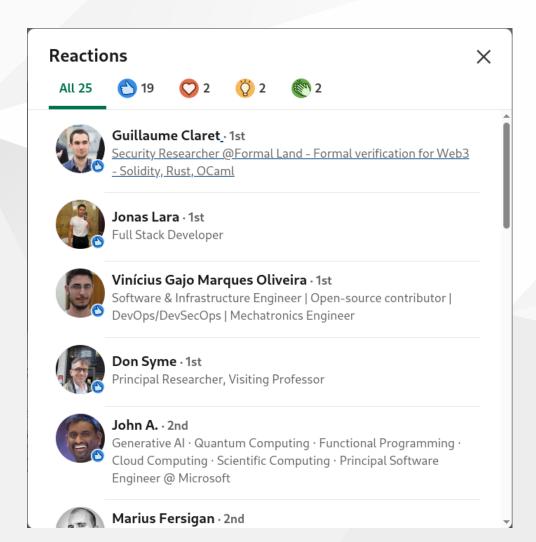
A bridge

- F# and .NET
- Simpler yet powerful language
- A work in progress

```
1  let ``double negation`` =
2    proof () {
3      Theorem("double negation", !(!x) ≡ x)
4      withLaws { ``true theorem`` }
5      !(!x) ≡ x
6      ``≡`` { ``GS 3.11`` }
7      !x ≡ !x
8      ``≡`` { ``≡ ident`` }
9      True
10   }
```
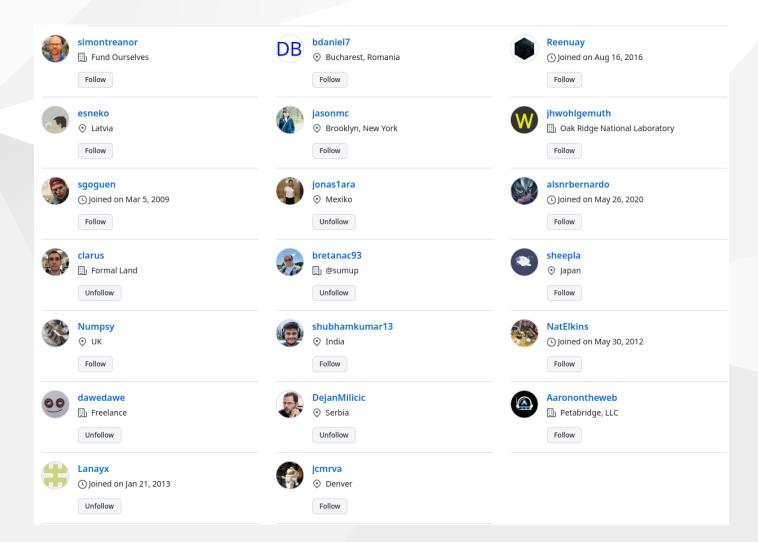
Reactions

8

Reactions

9

Reactions

Questions?

- https://github.com/lamg