Syntactic meta-programming in F#

Tomas Petricek, University of Cambridge
F# design principles (inferred)

Keep the **language simple**
- Syntax cannot be extended
- Syntax can be re-used

Follow the **80:20 rule**
- Use case: Write code for SQL, GPU, JavaScript
- Use case: Access external data sources
Kinds of meta-programming

1. Inp: Code
   Out: Data

2. In: Code
   Out: Code

3. In: Data
   Out: Data

4. In: Data
   Out: Code
Kinds of meta-programming

- F# Quotations
- N/A
- ?
- F# Type Providers
Two approaches in F#

Code from data

Earlier today. F# Type providers

Data from code

D. Syme, et al. Leveraging .NET metaprogramming components from F#
let [<ReflectedDefinition>] updateLoop () = async {
  while true do
    do! Async.Sleep(500)
    let! rects = getRectangles()
    drawRectangles(rects) 
}
Data from code