
# CS162 Final Review: Programming Languages

Yu by Yu Feng

# OCaml Programming Essentials

### Values

Immutable data primitives

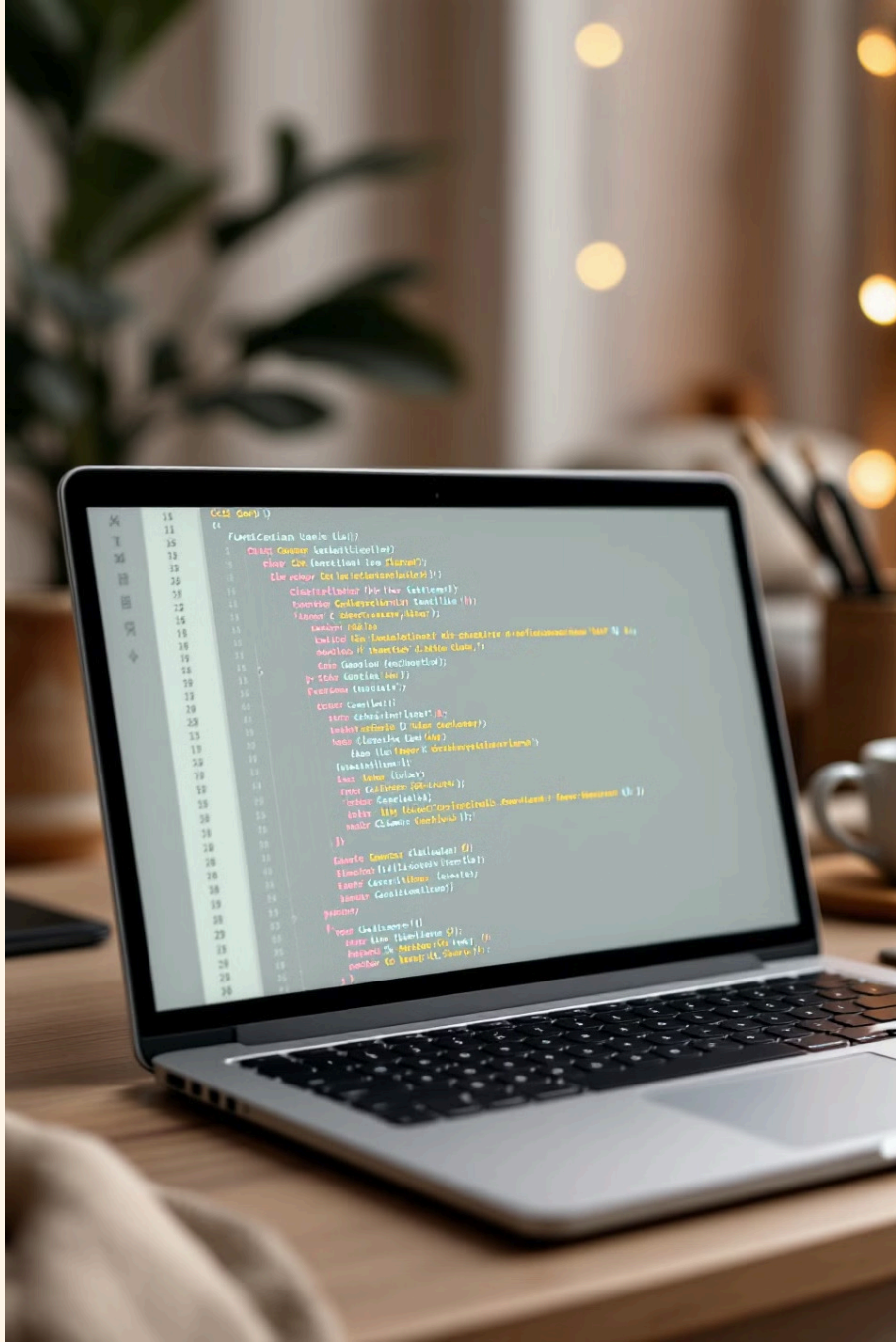
### Datatypes

Sum, product types, records

### Expressions

Everything evaluates to a value

### Pattern matching, recursion, higher-order functions



# Lambda Calculus Foundations

## Alpha Renaming

Rename bound variables without changing meaning

## Beta Reduction

Substitute argument into function body

## Evaluation Order

Call-by-value vs call-by-name

# Operational Semantics

1

## Big-Step

Directly relates expressions to final values

2

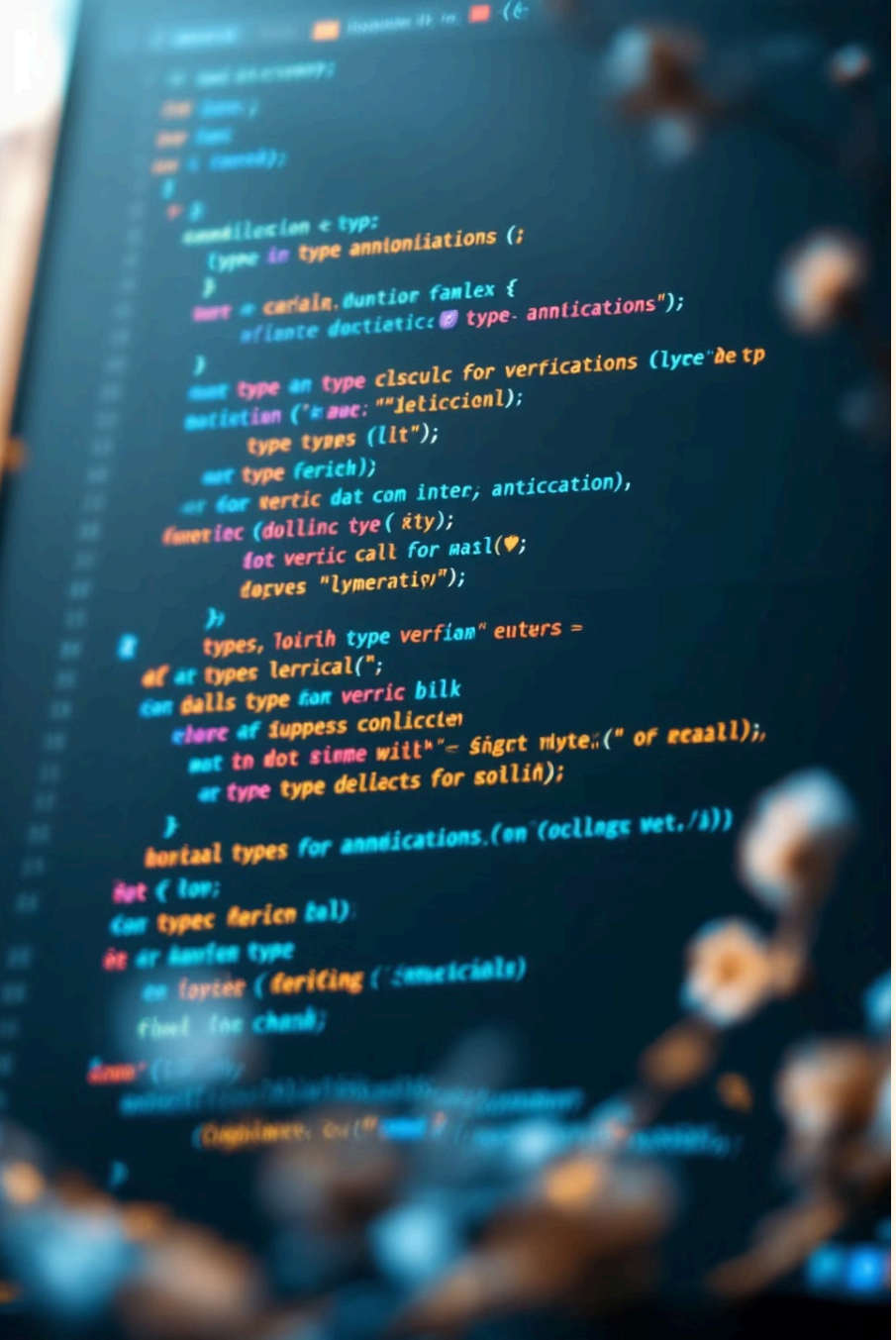
## Reduction Rules

Formal descriptions of evaluation steps

3

## Derivation Trees

Proof structures for expression evaluation



# Type Checking Principles

## Static Analysis

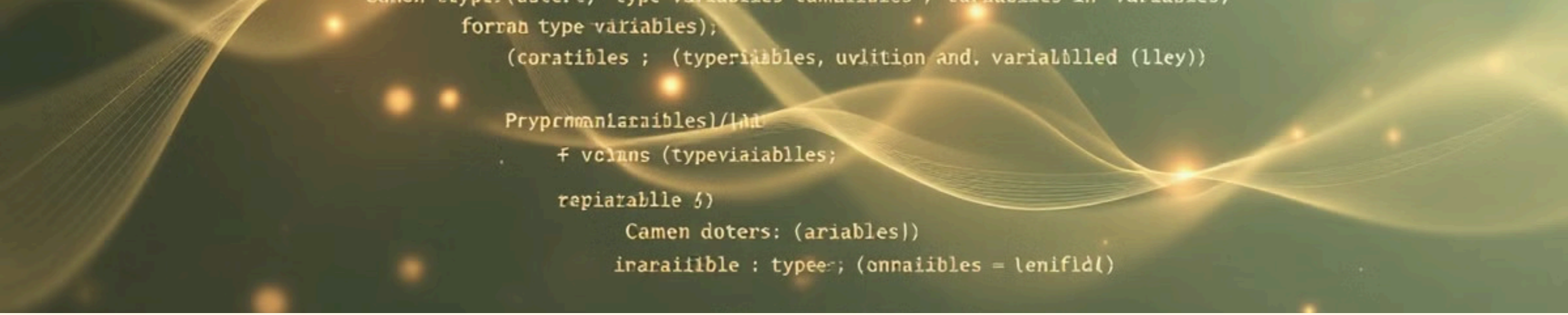
Verify program properties before execution

## Type Rules

Formal judgments about expression types

## Type Environments

Track variables and their types




# Type Inference Algorithms

## Constraint Generation

Collect equations based on program structure

## Unification

Solve equations to find most general types

## Substitution

Apply solutions to type variables

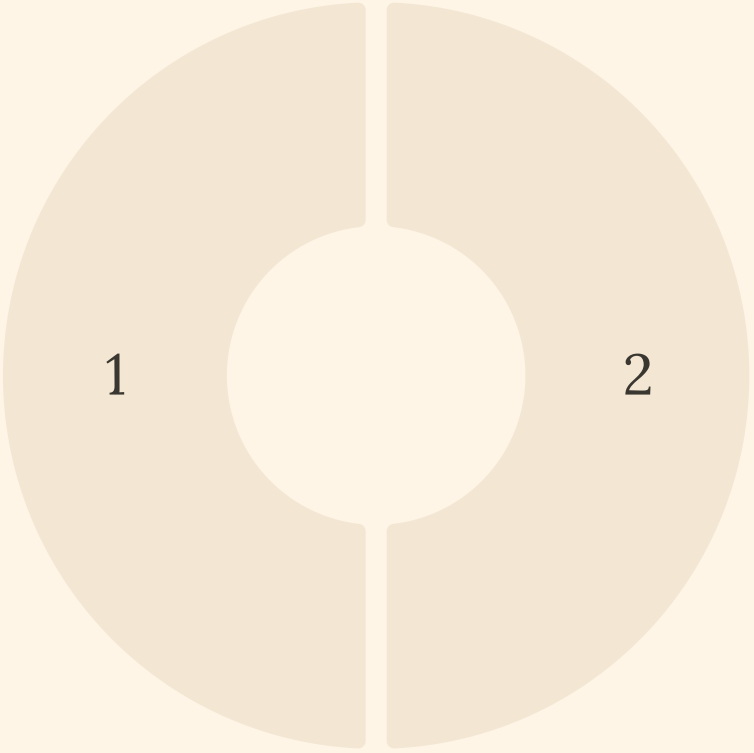
## Algorithm W

Hindley-Milner inference approach

# Polymorphism Systems

### Parametric

Works with any type




### Ad-hoc

Different implementations per type

# Putting It All Together

1. **OCaml Code**
   Write functional programs
2. **Lambda Basics**
   Understand theoretical foundation
3. **Semantics**
   Formalize evaluation rules
4. **Type Systems**
   Ensure program correctness