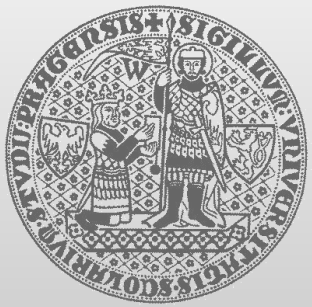
Clojure

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

Michal Malohlava

michal.malohlava@d3s.mff.cuni.cz



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

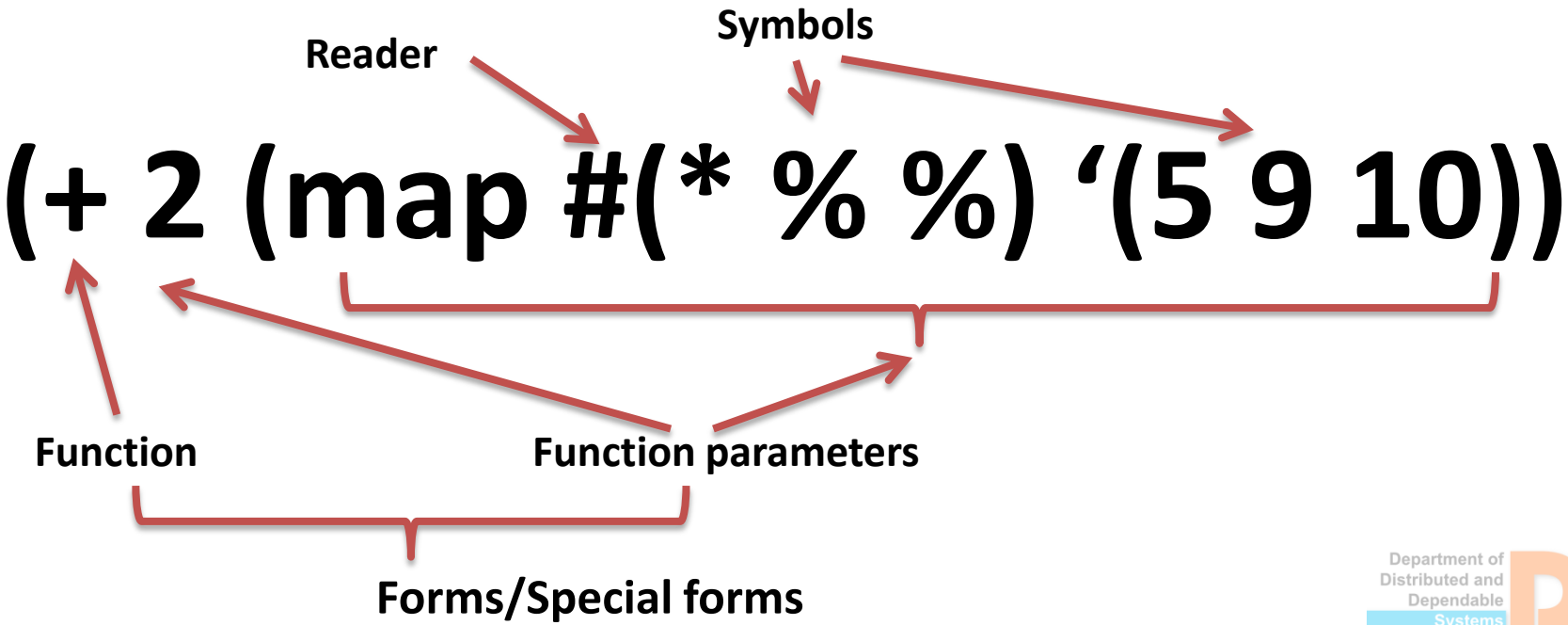
Intro

- **Clojure is LISP (LISP-1)**
  - LISP-1 = Data and code share the same namespace
    - I.e., Name resolving schema
  - LISP-2 = Scheme, Common Lisp
- Clojure is **functional language**
- **Run on JVM**
  - Interoperable with existing libraries

Department of Distributed and Dependable Systems D3S

Clojure syntax

- Syntax is based on **symbolic expressions** (s-expressions)
  - based on lists
    - (a (…) (b (…)))



Core concepts

- **Functional programming**
  - Immutable data
  - First-class functions
    - Higher-order functions
  - Function-based control flow
    - If/else, case are functions
  - Functions without side effects

Core concepts

- **Homoiconicity**
  - **Wikipedia says:**

"Homoiconicity is a property of some programming languages, in which the primary representation of programs is also a data structure in a primitive type of the language itself, from the Greek words homo meaning the same and icon meaning representation. This makes meta-programming easier than in a language without this property."

Core concepts

- **Code-as-data**
  - It is possible to manipulate with code thanks to
    - Homoiconicity
    - S-expressions
  - Extending Clojure via macros

IDE introduction

- **Clooj**
  - Written in Clojure
- Important shortcuts
  - *Tab* – shows help
  - *Ctrl+k* – clean the output
  - *Ctrl+r* – restart clojure
  - *Ctrl+e* – evaluate all commands in editor
  - *Ctrl+enter* – evaluate the current line2
- Project
  - Project.clj
    - Name, description, dependencies
    - Leiningen

REPL

- Clooj contains **REPL**
  - REPL = read-eval-print loop
- A way how to easily test statements and expressions of a language
- You can find REPL consoles in Scala, Clojure, ClojureScript, CoffeScript, Lua, …

Examples

- Basics
  - Functions
    - Lambdas,
  - Collections
    - list, map, set
- Meta programming
  - Multi-methods
  - Hierarchies
  - Macros
- DSL

DSL

- Have to follow syntax of Clojure
  - s-expression
- But you have
  - Functions with optional parameters
  - Macros
  - Code as data
  - HOF