LUNDS UNIVERSITET

Julia Course - Lecture 2

Mattias Fält

Lecture 2: Types, Functions and Multiple Dispatch

- Scopes
- Types
  - Type Tree
    - Abstract types
    - Concrete types
    - Parametric types
  - Defining
- Functions
  - Arguments
    - Optional argumenets
    - Keyword arguments
    - Typing
  - Operators With Special Names
  - As variables
    - Examples: sum, sort,...
    - Example odd/even
- Broadcasting
- Multiple dispatch
  - Some examples

Scoping

Script: `scopes.jl`

Types

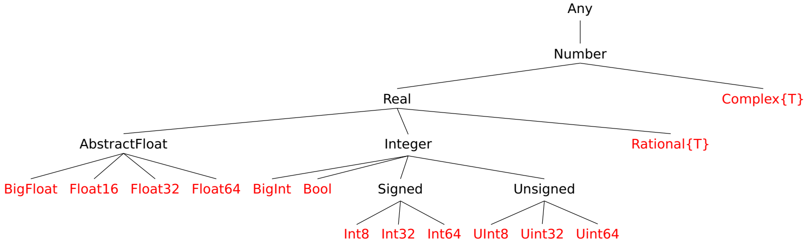

Figure: Part of Julia type tree

Types

Script: `types.jl`

Functions

Jupyter notebook: `functions.ipynb`

For code or instructions on jupyter, see: `functions.jl`

Dispatch

Script (Thanks to Stefan Karpinski): `multiple_dispatch.ipynb`

Dispatch

Example by Stefan Karpinski: `multiple_dispatch2.jl`

```
function innersum(A, vs)
    t = zero(eltype(A))
    for v in vs
        t += inner(v,A,v)
    end
    return t
end

inner(v, A, w) = dot(v, A*w)
```

So what happens in `innersum`?

```
inner(v, A, w) = dot(v, A*w)
```

- `A*w` calls generic matrix multiplication implementation
  - iterate through colums of `A` and multiplies them by each entry in `w`
  - returns a copy of column of `A` with type `Vector{Float64}`
- `dot(v,A*v)` calles generic dot implementation
  - does indexing into `v::OneHotVector` and `A*v::Vector{Float64}`

We can do much better based on our knowledge of OneHotVector!

Notes on multiple dispatch

- We are able to specialize on any or all of the arguments
- We can use types defined by anyone in our code!
- Our types works with other generic code without a problem!