

# Curious OCaml

Lukasz Stafiniak

## Curious OCaml

### From logic rules to programming constructs

What logical connectives do you know?

$\top$	$\perp$	$\wedge$	$\vee$	$\rightarrow$
truth	falsehood	$a \wedge b$	$a \vee b$	$a \rightarrow b$
“trivial”	“impossible”	conjunction	disjunction	implication
	shouldn’t get	$a$ and $b$	$a$ or $b$	$a$ gives $b$
		got both	got at least one	given $a$ , we get $b$

How can we define them? Think in terms of *derivation trees*:

$$\frac{\frac{\text{a premise} \quad \text{another premise}}{\text{some fact}} \quad \frac{\text{a premise} \quad \text{another premise}}{\text{another fact}}}{\text{final conclusion}}$$

To define the connectives, we provide rules for using them: for example, a rule  $\frac{a \ b}{c}$  matches parts of the tree that have two premises, represented by variables  $a$  and  $b$ , and have any conclusion, represented by variable  $c$ .

### Rules for Logical Connectives

Introduction rules say how to produce a connective. Elimination rules say how to use it. Text in parentheses is comments. Letters are variables: stand for anything.

Try to use only the connective you define in its definition.

TODO