# Curious OCaml

## Curious OCaml

## From logic rules to programming constructs

What logical connectives do you know?

| $\top$ | $\bot$ | $\wedge$ | $\vee$ | $\rightarrow$ |
|---|---|---|---|---|
| | | $a \wedge b$ | $a \vee b$ | $a \rightarrow b$ |
| truth | falsehood | conjunction | disjunction | implication |
| "trivial" | "impossible" | $a$ and $b$ | $a$ or $b$ | $a$ gives $b$ |
| | shouldn't get | got both | got at least one | given $a$, we get $b$ |

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

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

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