Declarations may give the type of an identifier, but typing rules establish how expressions of various types are composed.

... computer science, not set theory.

They are algorithms mapping values of one type to values of another type.

Reducing terms until they cannot be further reduced.

The *strong normalization* property is too strong. Any system that can describe all computable functions must also admit functions that don't halt. Since the strong normalization property guarantees all Coq functions halt, the ability to compute some computable functions is necessarily left out.

... terms ... a type.

## `Prop`

Using an arrow type ending in `Prop`, with the earlier types being the types of the variables in the proposition.

Propositions can refer to programs (e.g., the `Prop` type), and programs can refer to propositions (*dependent types*).

As in OCaml:

```
(* comment *)
```