The Terminal Function & Propositions of Irrelevance

by Sven Nilsen, 2017-2019

Assume you have the following functions:

```
human : object → bool
mortal : object → bool
mortal([human] true) = true
```

All that is needed to determine whether a human is mortal, is knowing “the object is human”.
No other information is required.

Assume you make a computer program where a person has a name and age:

```
objs := [
                {name: “John”, age: 32},
                {name: “Peter”, age: 48},
                {name: “Carl”, age: 89},
        ]
```

You call a function “is_mortal” like this:

```
is_mortal(objs[i].name, objs[i].age)

is_mortal : string × f64 → bool
```

Since all objects are humans, this function will always return `true`.
How do you prove that the object’s name and age is irrelevant to the result of “mortal”?

The Terminal function (often called “unit”) can be used to remove information in a path:

```
unit(a) → ()
unit(_) = ()
```

Now you can write:

```
is_mortal[unit × unit → id] <=> \((), ()) = true
```

This is a proposition of the irrelevance of the two arguments.