Hello, `noweb`!

An exploration of literate Idris programming via `noweb`.

Eric Bailey

January 14, 2017

Abstract

What follows is an attempt at using `noweb` to write a literate program in Idris. While Idris provides some literate programming support of its own, it's rather basic (like Literate Haskell), and doesn't allow users to present code chunks out of order or do any sort of cross-referencing.

Contents

1 Module Declaration

1 ⟨*hello* 1⟩≡ 4▷
⟨*Module declaration* 2⟩

First, we declare the `Main` module, including a docstring, which is another `noweb` chunk, because we can.

2 ⟨*Module declaration* 2⟩≡ (1)
```
||| ⟨Hello message 3⟩
module Main
```

The docstring above[1] consists of the following message, which we'll also print later, using another `noweb` reference.

3 ⟨*Hello message* 3⟩≡ (2 6)
```
Hello, noweb!
```

2 The `main` Function

4 ⟨*hello* 1⟩+≡ ◁1
⟨*main type signature* 5⟩
⟨*main definition* 6⟩

`main` is an I/O action that doesn't return any value, i.e.

5 ⟨*main type signature* 5⟩≡ (4)
```
main : IO ()
```
Defines:
`main`, used in chunk 6.

The action, when run, will output the message to `stdout` with a trailing newline.

6 ⟨*main definition* 6⟩≡ (4)
```
main = putStrLn "⟨Hello message 3⟩"
```
Uses `main` 5.

[1] The lines of an Idris docstring start with `|||`, and immediately precede the definition they document.

3 Chunks

⟨*hello* 1⟩
⟨*Hello message* 3⟩
⟨*main definition* 6⟩
⟨*main type signature* 5⟩
⟨*Module declaration* 2⟩

4 Index