

## NFAStream

### Overview:

NFAStream is a project written in Haskell and intended for use with the ReWire project. (see <http://muchaco.github.io/ReWire/>). ReWire compiles Haskell programs into VHDL in order to build FPGA or integrated circuit systems with formally verified program properties. NFAStream lets a user define a pattern by means of a regular expression and filter a stream of data for any instance of the pattern. Such a program compiled down to VHDL is a rudimentary example of network traffic filter.

### Usage:

NFAStream can be used with any interactive Haskell environment such as GHCi from the Haskell Platform. Unfortunately NFAStream is a technical demonstration and has no user friendly interface. In order to set up a filter you must build a Haskell algebraic data type representing a regular expression in Polish notation. Use the runState command from Haskell's State Monad library in order to check a stream against a regular expression. For example, to check the stream 'Hello' against the regular expression 'n|el\*' enter this command:

```
runState (stream 'H' >> stream 'e' >> stream 'l' >> stream 'l' >> stream 'o') (r2s ( Bar (Sym 'n') ( Sym 'e' )  
:| Star ( Sym 'l' )))
```