Regular Expression Interpreter Implementation

Usage and overview

The program is implemented as a command line application and is written in Rust. To run it, you just need to install Rust, clone the project and build it using

```
cargo build
```

or

```
cargo run
```

in the project folder. The latter one will also print usage instructions. For normal usage, the program takes a regular expression as the first and only argument like so:

```
rs-regex "a(b|cc)*a"
```

or

```
cargo run "a(b|cc)*a"
```

After that, the program will ask strings as inputs one by one and tell you whether they belong to the language (defined by the regex) or not. An empty string will exit the program.

Alternatively, if you want to see how the regular expression is tokenized, you can give the option -t as on optional argument along with the regex. The program will print all of the tokens along with their values and exit. The output looks like this:

```
rs-regex "a(b|c)*" -t

Token(Char, a)
Token(LeftParen, ()
Token(Char, b)
Token(Union, |)
Token(Char, c)
Token(RightParen, ))
Token(Star, *)
```

The regular expression should be given inside double quotes. The set of supported characters are ASCII (8-bit), from which some special ones, such as the operator symbols, need to be escaped using backslash, e.g. *. The supported operators are:

Operator	Syntax	Matches
Union	A | B	"A" or "B"

Star	a*	0 or more "a"
Concatenation	01	"0" followed by "1"
Group	(a|bb)*	0 or more "a" or "bb"

The overview of the implementation looks like this: