Flex/JFlex Cheat Sheet

Files structure

Flex

```
Definitions
%%
Lexical Rules
%%
User Code
```

In `Definitions` are macros for regular expressions, states, options, and C code (enclosed in %{...}%) to be copied at the top of the file.
`Lexical Rules` are patterns associated to actions.
`UserCode` is where functions used in `Lexical Rules` are defined, is copied verbatim, and is optional. `main` can be defined in it.

JFlex

```
UserCode
%%
Options and
Declarations
%%
Lexical Rules
```

`UserCode` is copied verbatim to the generated file.
`Options and Declarations` set global options and allow to define custom variables, methods and states. Useful options includes `%line`, `%column`, `%standalone`
`Lexical Rules` contains patterns associated to actions.

Patterns

In the following tables, r, s are regular expressions. Note that the semantics of these pattern might subtly change depending on the options, the encoding or the platform used. Whenever in doubt, refer to the manuals of Flex and JFlex.

Common

Flex/JFlex	Matches
`x`	the character 'x'
`.`	any character except `\n`
`\n`	newline. Better : `\r|\n|\r\n`.
`[xyz]`	a character class, matches 'x', 'y', 'z'.
`[abj-oZ]`	a character class, matches 'a', 'b', 'j' through 'o' (lowercase) and 'Z'.
`[^A-Zk]`	negated character class. Any character except those in uppercase or 'k'.
`"[xyz]\"foo"`	the literal string '[xyz]"foo'
`\0`	NULL character.
`\*`	character '*'. Works for any character except 'a', 'b', 'f', 'n', 'r', 't', 'v', '0'.
r`*`	zero or more r's.
r`+`	one or more r's.
r`?`	zero or one r's.
r`{2, 7}`	from two to seven r's.
r`{3,}`	three or more r's.
r`{4}`	four r's.
(r)`{4}`	r. Override precedence.
rs	r followed by s.
r/s	r followed by s (trailing context). Returned matched text is r only. s can be rematched.
$r\|s$	r or s.
`^`r	r at the begining of a line
r`$`	r at the end of a line
`<`s`>`r	r when in state s.
`<`s_1`, `s_2`, `s_3`>`r	r when in state s_1, s_2 or s_3.
`<<EOF>>`	end of file.
{NAME}	expand the macro with name 'NAME'

Flex

Flex is sensitive to whitespaces and tabs in patterns.

Flex	Matches
`[a-e]-[bd]`	character 'a', 'c', 'e'.
`[a-c]+[gh]`	character 'a', 'b', 'c', 'g', 'h'.
rs	r followed by s (concatenation).
`<*>`r	r when any state (even exclusive).
`[:alnum:]`	predefined character class, equivalent to the standart C `isalnum()` function. Other class are `alpha`, `blank`, `cntrl`, `digit`, `graph`, `lower`, `print`, `punct`, `space`, `upper`, `xdigit`.
`(?# comment )`	match everything (including `\n`) between '(' and ')'.

JFlex

JFlex allows, and ignores, whitespaces and tabs in patterns, except in character classes and strings. This can be used to improve pattern readibility

Flex	Matches
$r\ \ s$	r followed by s (concatenation).
`!`r	negation. Everything not r. Be careful about an exponential NFA to DFA transformation when using negation.
`~`r	everything up to the first occurence of r.
`\n`	For unicode compliance, add : `\u2028|\u2029|\u000B|\u000C|\u0085`
`[:jletter:]`	predefined character class, equivalent to `java.lang.Character.isJavaIdentifierStart()`. Other class are `jletterdigit`, `letter`, `digit`, `uppercase`, `lowercase`.

v0.92, by Fabien Benureau (fabien.benureau@inria.fr). Corrections and comments welcome.