Task documentation SYN: Syntax highlighting in Python for IPP 2013/2014
Name and surname: Ján Spišiak
Login: xspisi03

1 Task Description

Write a script in Python which wraps expressions in input file with tags, defined by optional formatting file, and outputs the result. The formatting file consists of custom regular expressions and their formatting.

2 Implementation

2.1 Parsing arguments

Using python's own `argparse` we simply define set of accepted arguments. We also have to check if some argument wasn't specified twice, `argparse` doesn't do this for us, so we simply check the original `sys.argv`.

2.2 Parsing formatting

Parsing strings in python is stupendously easy. We split the line into regex and the format options separated by first "`\t+`". Format options are easily separated by "`\s*,\s*`", and then forwarded to function `formatToTag` which checks option validity and also filters additional data. The regex is fed into `parseRe` which converts IFJ RE into Python RE using state machine. State machine has 5 states: expr, any, negation, negationEscape and escape. The NQS extension is also implemented here, using `quant` variable remembering last repeat operator, but the same effect could also be achieved using states, although with more lines of code. Tuple (REGEX, (OPEN_TAGS, CLOSE_TAGS)) is then saved in `formatting` list.

2.3 Matching expressions

We iterate over the matches for each regex and save it's number (in `formatting` list) at match start/end into `positions` dictionary. It's important to use `re.DOTALL` option, otherwise `.` won't match newline as in specification.

2.4 Merging result

Merging `positions` dict and `source` is relatively easy. Insert anything between last position and current one, then insert end and open tags (which we need to lookup in formatting dictionary). This method leaves open possibility of implementing HTM extension with simple stack. Append end of the source which we might not have inserted and that's it.

3 Conclusion

The program works as intended, given source document it applies tags to expressions as specified in formatting file. The priority is also preserved, although the HTM extension wasn't implemented so possible crossover tag structure is possible.