

Лабораторная работа № 1.2. «Лексический анализатор на основе регулярных выражений»

26 февраля 2024 г.

Сергей Виленский, ИУ9-62Б

Цель работы

Целью данной работы является приобретение навыка разработки простейших лексических анализаторов, работающих на основе поиска в тексте по образцу, заданному регулярным выражением.

Индивидуальный вариант

Географические координаты: начинаются с одного из знаков «S», «E», «N», «W», после которых располагается целое десятичное число, за которым может следовать либо точка и последовательность десятичных цифр, либо знак «D», за которым следует необязательная запись угловых минут (число от 0 до 59, за которым пишется апостроф) и угловых секунд (число от 0 до 59, за которым следует двойная кавычка). Атрибут лексемы (для лабораторных работ 1.3 и 1.5): вещественное число, соответствующее широте или долготе. Широта («S», «N») не может превышать 90, долгота («E», «W») — 180.

Реализация

```
import sys
from typing import Generator
import re

RANGE_0_59 = r"0*[1-5]?\\d" # range(0, 60)
ZERO_MM_SS = \
    r"0+" r"\\s*" \
    r"" r"\\s*" \
    r"0+" r"\\s*" \
    r"" r"\\s*" \
```

```

ANY_MM_SS =
    fr"{RANGE_0_59}"    r"\S*" \
    r" "                r"\S*" \
    fr"{RANGE_0_59}"    r"\S*" \
    r"\ "               r"\S*"

ZERO_DDDD = r"0*"    r"\S*"
ANY_DDDD = r"d*"    r"\S*"

ZERO_END = fr"(\.{ZERO_DDDD}|\s*D\s*{ZERO_MM_SS})"
ANY_END = fr"(\.{ANY_DDDD}|\s*D\s*{ANY_MM_SS})"

RANGE_0_89 = r"[1-8]?d"
RANGE_90_90 = r"90"

RANGE_00_79 = r"[0-7]d"
RANGE_0_99 = r"[1-9]?d"
RANGE_0_179 = fr"(1{RANGE_00_79}|{RANGE_0_99})"
RANGE_180_180 = r"180"

LATITUDE = r"\S*" r"0*" \
    fr"({RANGE_0_89}{ANY_END})" \
    fr"{RANGE_90_90}{ZERO_END})"
LONGITUDE = r"\S*" r"0*" \
    fr"({RANGE_0_179}{ANY_END})" \
    fr"{RANGE_180_180}{ZERO_END})"

PATTERN_GEOGR_COORDS = r"\S*" \
    fr"([SN]{LATITUDE})" \
    fr"([EW]{LONGITUDE}|)" \
    r"\S*"

def lexems_iter(
    source_code: str
) -> Generator[tuple[bool, str, int, int, str], None, None]:

    pattern = re.compile(PATTERN_GEOGR_COORDS, re.UNICODE)

    for line_index, file_line in enumerate(source_code.split('\n'), start=1):
        last_lexem_end = 0

        for next_iter in pattern.finditer(file_line):
            lexem_col_from, lexem_col_to = next_iter.span()
            if lexem_col_from == lexem_col_to:
                continue

```

```

        if last_lexem_end != lexem_col_from:
            yield True, "", line_index, last_lexem_end, ""

        yield False, "COORDS", line_index, lexem_col_from, next_iter[0]
        last_lexem_end = lexem_col_to

    if last_lexem_end != len(file_line):
        yield True, "", line_index, last_lexem_end, ""

with open(sys.argv[1], mode="r", encoding="utf8") as input_file:

    for (
        lexem_error,
        lexem_type,
        lexem_line,
        lexem_col,
        lexem_str
    ) in lexems_iter(input_file.read()):
        if lexem_error:
            print(f"syntax error ({lexem_line}, {lexem_col})")
        else:
            print(f"{lexem_type} ({lexem_line}, {lexem_col}):",
                  repr(lexem_str))

```

Тестирование

Входные данные

```

S 0000.
S 0000.000
S 0089.999
S 0090.000
S 0090.
S0090.001
S0091.000

E 0000.
E 0000.000
E 0179.999
E 0180.000
E 0180.
E0180.001
E0181.000

```

```

S 0000D'000"000
S 0089D'059"059
S 0090D'000"000
S0089D'060"059
S0089D'059"060
S0090D'001"000
S0090D'000"001

```

```

E 0000D'000"000
E 0089D'059"059
E 0090D'000"000
E0089D'060"059
E0089D'059"060
E0090D'001"000
E0090D'000"001

```

Вывод на stdout (если необходимо)

```

COORDS (1, 0): ' S 0000.          '
COORDS (2, 0): ' S 0000.000      '
COORDS (3, 0): ' S 0089.999      '
COORDS (4, 0): ' S 0090.000      '
COORDS (5, 0): ' S 0090.          '
COORDS (6, 0): 'S0090.00'
syntax error (6, 8)
syntax error (7, 0)
COORDS (9, 0): ' E 0000.          '
COORDS (10, 0): ' E 0000.000     '
COORDS (11, 0): ' E 0179.999     '
COORDS (12, 0): ' E 0180.000     '
COORDS (13, 0): ' E 0180.          '
COORDS (14, 0): 'E0180.00'
syntax error (14, 8)
syntax error (15, 0)
COORDS (17, 0): ' '
syntax error (17, 1)
COORDS (17, 2): ' '
syntax error (17, 3)
COORDS (17, 16): '          '
COORDS (18, 0): ' '
syntax error (18, 1)
COORDS (18, 2): ' '
syntax error (18, 3)
COORDS (18, 16): '          '
COORDS (19, 0): ' '
syntax error (19, 1)
COORDS (19, 2): ' '

```

```
syntax error (19, 3)
COORDS (19, 16): ' '
syntax error (20, 0)
syntax error (21, 0)
syntax error (22, 0)
syntax error (23, 0)
COORDS (25, 0): ' '
syntax error (25, 1)
COORDS (25, 2): ' '
syntax error (25, 3)
COORDS (25, 16): ' '
COORDS (26, 0): ' '
syntax error (26, 1)
COORDS (26, 2): ' '
syntax error (26, 3)
COORDS (26, 16): ' '
COORDS (27, 0): ' '
syntax error (27, 1)
COORDS (27, 2): ' '
syntax error (27, 3)
COORDS (27, 16): ' '
syntax error (28, 0)
syntax error (29, 0)
syntax error (30, 0)
syntax error (31, 0)
```

Вывод

В результате выполнения данной работы был приобретен навык разработки простейших лексических анализаторов, работающих на основе поиска в тексте по образцу, заданному регулярным выражением.