

# Пояснение к практикуму №1

Ахтямов Павел, 399 группа

October 22, 2014

Суть алгоритма, который был предложен на практикум (проверить, лежит ли подслово  $x^k$  в некотором слове языка, распознаваемого регулярным языком), состоит в динамическом программировании по стеку, который задается польской нотацией. Мы будем хранить в стеке 4 параметра:

*mostLengthPrefix\_* — длина наибольшего префикса, который состоит из букв  $x$ , слов, распознаваемых соответствующим регулярным выражением.

*mostLengthSuffix\_* — длина наибольшего суффикса, который состоит из букв  $x$ , слов, распознаваемых соответствующим регулярным выражением.

*mayFull\_* — может ли быть слово, распознаваемое данным регулярным выражением, состоять только из букв  $x$

*maxFullLength\_* — если *mayFull\_* = *true*, то длина наибольшего такого слова, иначе — 0.

При этом полагаем, что значения могут быть равными  $+\infty$ , что реализовано в классе *IntWithInfinity* (будем считать, что  $+\infty > +\infty$ ).

При этом будем поддерживать длину наибольшего количества букв  $x$  в слове в параметре *maximalLength*.

Для букв и 1 параметры задаются естественным образом. Далее, для объединения мы должны выбирать максимальные из параметров, чтобы потом их использовать в дальнейшем.

С итерацией можно получить слово с бесконечным количеством  $x$  подряд, для этого необходимо и достаточно, чтобы это слово состояло только из букв  $x$ , и, причем их количество было больше 0, иначе можно получить только слово, длина которого равна сумме значения для префикса и для суффикса, ибо мы получаем конкатенацию нескольких слов, которые распознавались, при этом нет *mayFull*. Тогда можно получить новую максимальную длину только приписыванием одного слова к другому. В конце надо поставить *mayFull\_* = *true* и проставить соответствующую длину: либо  $+\infty$ , либо 0 (пустое слово тоже состоит из одних букв  $x$ ).

Для конкатенации возможны различные случаи: для *mayFull\_* — два слова должны быть *mayFull\_*, иначе их конкатенация точно не будет содержать букв  $x$  (и их длина будет равна сумме длин). Как обновлять длины максимальных префиксов и суффиксов: для префикса необходимо проверить первое слово на полноту, а потом прибавить префикс второго, для суффикса — наоборот, и, естественно, случаи префикса первого и суффикса второго слова никто не отменял (поэтому при наличии нескольких возможностей выбираем максимум). Максимальная длина может увеличиться в случае, когда мы берем суффикс первого слова и конкатенируем префикс второго слова (при этом  $maxFullLength \leq mostLengthPrefix$  и  $maxFullLength \leq mostLengthSuffix$  и мы их обновляем соответствующим образом).

В программе перед запуском происходит запуск на unit-тестах, которые покрывают некоторые случаи, в которых могут быть допущены ошибки.

К программе приложен Makefile, команда для компиляции

```
make all clean
```

Исполняемый файл, который появляется при сборке — *regex.exe*.

Данный код доступен по репозиторию <https://github.com/akhtyamovpavel/TranslatorPractice>