Теория к задаче Нечеткий поиск

1 Алгоритм решения

Получаем на вход две строки: *pattern* и *text*.

Если размер *pattern* больше *text* или одна из строчек пустая, возвращаем пустой массив.

Пусть символы *wildcard* встречаются в строке *pattern* на позициях $wildcard\_pos_i$, $i = 1 \dots \#wildcards$. Тогда разбиваем *pattern* по *wildcard* символам на подстрочки, и составляем из них массив *subpatterns*, $subpatterns[i] = pattern[begin_i : begin_{i+1} - 1)$, где $begin_0 = 0, begin_i = wildcard\_pos_i, begin_{\#wildcards+1} = pattern.size + 1$.

Заводим массив *candidates* размера *text.size*, в i-ой ячейке которого лежит массив булевых флагов размера $subpatterns.size = \#wildcards + 1$.

Запускаем алгоритм Ахо-Корасик, ищя подстрочки *subpatterns* в *text*. По мере увеличения размера обработанного префикса $text[0 : i]$, алгоритм Ахо-Корасик будет возвращать нам множество индексов $\{j\}$, соответствующих строкам массива *subpatterns*, которые встречаются в тексте и заканчиваются на позиции i. Тогда, начиная с позиции $i = pattern.size$, мы выставляем соответвующие флаги $candidates[k_j][j]$, где k_j - такая позиция в *text*, что если бы там начинался *pattern*, то мы бы увидели строчку $subpattern_j$ в этом же месте, то есть $k_j = i - pattern.size + begin_j + 1$.

После этого проходим по массиву *candidates*, и выписываем все индексы i, такие, что все $\#wildcards + 1$ флагов $candidates[i]$ отмечены.

2 Доказательство правильности алгоритма

Если размер *pattern* больше *text* или одна из строчек пустая, то вхождений быть не может.

Иначе если в *text* есть вхождение *pattern* на месте i, то алгоритм его найдет. Если бы i не было выписано, то в $candidates[i]$ не отмечен какой-то флаг. Но это значит, что на ожидаемом месте не нашлась подстрока без *wildcard* символов, что противоречит корректности алгоритма Ахо-Корасик.

И обратно, если алгоритм выдал индекс i, то там вхождение действительно есть, поскольку это означает, что все флаги в $candidates[i]$ отмечены, и все символы в $text[i : i + pattern.size)$, отличные от *wildcard*, совпадают с соответсвующими в *pattern*.

3 Временная сложность — асимптотика

Нахождение позиций $wildcard\_pos_i$ и построение массива $subpatterns$ - $O(pattern)$.

Создание массива $candidates$ - $O(text * \#wildcards)$

Алгоритм Ахо-Корасик и заполнение массива $candidates$ - $O(text+pattern+\#occurrences) = O(text * \#wildcards)$.

Проход по массиву $candidates$ и выписывание ответов - $O(text)$.

Общая сложность - $O(text * \#wildcards)$.

4 Затраты памяти — асимптотика

Для массива $subpatterns$ – $O(\#wildcards)$

Для массива $candidates$ - $O(text * \#wildcards)$

Для алгоритма Ахо-Корасик - $O(pattern)$

Для ответа - $O(text)$

Общие затраты - $O(text * \#wildcards)$.