

Теория формальных языков

Практикум №1

Регулярные выражения

Константин Чернис, группа 694

Описание алгоритма (вариант 10)

Дополнительные обозначения:

- x — требуемый символ
- k — требуемая длина суффикса

Для каждого языка, порождаемого частью регулярного выражения, будем хранить следующую информацию:

1. **ends_with_letter**: словарь, в котором для каждой длины m суффикса вида x^m хранится минимальная длина слова, имеющая такой суффикс
2. **only_letter**: множество, в котором хранятся всевозможные длины слов вида x^m (ради эффективности туда не добавляются слишком длинные суффиксы)
3. **best_ends**: минимальная длина слова, содержащая суффикс x^k (если нет — то ∞ (очень большое число))
4. **best_only**: минимальная длина слова вида x^m для $m \geq k$ (если нет — то аналогично предыдущему пункту)

Тогда для получения ответа достаточно аккуратно обработать все 3 типа операций, после чего ответ будет в информации о языке, полученном последовательным применением всех операций.

Ответ получится корректным по индукции по длине регулярного выражения, где в коде программы задаётся база, а переход описан ниже.

Описание обработки операций

Операция $+$:

1. **best_ends** — минимум из значений **best_ends** для операндов
2. **best_only** аналогично **best_ends**

3. **only_letter**: объединяем множества **best_only** операндов (все длины возможны)
4. **ends_with_letter**: если данная длина x^m содержится только в одном из **ends_with_letter**, то оставляем, иначе берём минимум длин

Далее будем рассуждать в терминах ax (слова, оканчивающиеся на нужную букву) и xx (слова, состоящие только из нужной буквы).

Операция .:

1. К правому ax приклеиваем самое короткое слово из левого операнда и обновляем **best_ends**
2. К левому ax приклеиваем всевозможные слова из правого xx , соблюдая условие минимальности и обновляя **best_ends**. Таким образом, построили **ends_with_letter**
3. Аналогично склеиваем xx и xx , делая то же, что и в предыдущем пункте. Получаем **only_letter** и **best_only**

Операция *:

1. Добавляем в **best_ends** и **best_only** слова длины 0
2. Обновляем **only_letter** итеративно до тех пор, пока множество не изменится после добавления очередной длины (не добавляем слово, если есть более короткое, которое уже имеет требуемую длину)
3. Приклеиваем к ax всевозможные окончания из построенного **only_letter**, попутно обновляя **best_ends**, получаем **ends_with_letter**