Юсупов Никита, 694 группа

Практикум по формальным языкам

Задача №14

Дано слово W, найти подслово слова W максимальной длины, такое, что оно является подсловом некоторого слова из языка L.

Решение:

Перебем все подслова слова W, пусть мы зафиксировали какое-то подлсово U, проверим, является ли U подсловом некоторого слова в языке L. Выберем Umax — U с максимальной длиной из всех тех U, которые являются подсловом некоторого слова в языке L. Длина Umax и будет ответом на нашу задачу.

Итак, мы зафиксировали U, нужно проверить, является ли оно подсловом некоторого слова из языка L. Будем решать задачу с помощью индукции по построению. Для каждого языка будем хранить некоторые характеристики и вычислять их значения для языка L1.L2, L1+L2, (L1)*, зная значения этих характеристик для L1 и L2.

Опишем характеристики языка L:

1. containsSubstring[i][j] == true <=> подслово слова U длины j, начинающееся в i-ой позиции, содержится в языке L.
2. containsEpsilon == true <=> пустое слово принадлежит нашему языку L.
3. containsWordAsSubstring == true <=> слово U содержится в качестве подслова какого-либо слова из языка L (Главный параметр, который и является ответом на нашу подзадачу)
4. containsSuffixEqualsToPrefix[i] == true <=> в языке L есть слово V, такое, что суффикс (длины i) S слова V равен префиксу длины i слова U.
5. containsPrefixEqualsToSuffix[i] == true <=> в языке L есть слово V, такое, что префикс (длины i) P слова V равен суффиксу длины i слова U.

(Все эти характеристики еще раз описаны в коде в виде комментариев)

Опишем базу, то есть характеристики для языка, состоящего из одного слова длины <= 1, то есть это слово равно либо какой-то букве x, либо пустому слову.
Изначально все поля равны false.

1. containsSubstring[i][1] := true, если U[i] == x.
2. containsEpsilon := true, если x — пустое слово.
3. containsWordAsSubstring := true, если U == x.
4. containsSuffixEqualsToPrefix[1] := true, если U[0] == x.
5. containsPrefixEqualsToSuffix[1] := true, если U[U.length - 1] == x.

Опишем переходы, то есть сейчас мы знаем характеристики L1 и L2 и вычисляем характеристики языка (L1 + L2), (L1 . L2), (L1)*:

(L1 + L2):
Здесь все достаточно тривиально, если какая-либо из характеристик была равна true либо в языке L1, либо в языке L2, либо и там и там, то в языке (L1 + L2) эта характеристика будет равна true, а иначе false.

(L1 . L2):
(Комментарии и пояснения для этих переходов так же присутствуют в коде)
1. L.containsSubstring[i][j] := true, если
   - Либо найдутся такие prefixLength и suffixLength, что:
     prefixLength > 0 и
     suffixLength > 0 и
     prefixLength + suffixLength == j и
     L1.containsSubstring[i][prefixLength] == true и
     L2.containsSubstring[i + prefixLength][suffixLength] == true
   - Либо L1.containsEpsilon == true и L2.containsSubstring[i][j] == true
   - Либо L2.containsEpsilon == true и L1.containsSubstring[i][j] == true
2. L.containsEpsilon := true, если
   L1.containsEpsilon == true и
   L2.containsEpsilon == true
3. L.containsWordAsSubstring := true, если
   - Либо L1.containsWordAsSubstring == true
   - Либо L2.containsWordAsSubstring == true
   - Либо найдется такое число prefixLength > 0, что
     L1.containsSuffixEqualsToPrefix[prefixLength] == true и
     L2.containsPrefixEqualsToSuffix[U.length - prefixLength] == true

4. L.containsSuffixEqualsToPrefix[i] := true, если
   - Либо L2.containsSuffixEqualsToPrefix[i] == true
   - Либо L1.containsSuffixEqualsToPrefix[i] == true и L2.containsEpsilon == true
   - Либо найдется такое число subPrefixLength > 0, что L1.containsSuffixEqualsToPrefix[subPrefixLength] == true и L2.containsSubstring[subPrefixLength][i - subPrefixLength] == true
5. L.containsPrefixEqualsToSuffix[i] := true, если
   - Либо L1.containsPrefixEqualsToSuffix[i] == true
   - Либо L2.containsPrefixEqualsToSuffix[i] == true и L1.containsEpsilon == true
   - Либо найдется такое число subSuffixLength > 0, что L2.containsPrefixEqualsToSuffix[subSuffixLength] == true и L1.containsSubstring[U.length - i][i - subSuffixLength] == true

(L1)*:

$L^* == L^0 + L^1 + L^2 + \dots + L^n + L^{n+1} + \dots$

Будем честно вычислять значения L* для каждого шага. То есть, сначала вычислим L^0, потом $L^0 + L^1$, и так далее.
Понятно, что нам достаточно посчитать только 2*U.length шагов, так как уже на шаге с номером U.length у нас перестанут появляться новые префиксы слова U, так как все новые слова будут длины, большей, чем длина слова U, по той же причине у нас не будут появляться в языке новые подслова слова U, а также, заметим, что, если на шаге с номером N, большим, чем 2*U.length у нас появилось слово в языке с суффиксом, равным какому-либо префиксу слова U, то такое слово уже было в языке на шаге N/2, так как $L^N = L^{N/2} \cdot L^{N/2}$. Следовательно, значения характеристик для языка (L1)* мы будем вычислять, используя операции сложения и умножения, которые мы уже определили.

Таким образом, мы умеем вычислять значения для всех операций, а значит мы можем вычислить для языка, заданного исходным регулярным выражением, значение параметра containsWordAsSubstring, который и будет давать нам ответ на нашу подзадачу.