Formal-Practicum-1
Variant 1

Головацкий Андрей
Б05-925

9 November 2020

1 Условие

Даны α (регулярное выражение в обратной польской записи, задающее язык L) и натуральные числа k, l, такие что $0 \leq l < k$. Проверить, содержит ли язык L слова, чья длина равна l по модулю k.

2 Решение

Решим данную задачу, при условии того, что нам известен конечный детерминированный автомат(НКА) - A, эквивалентный выражению α. Отметим, что A может содержать ε-переходы

Пусть слово $w \in L, |w| = n$, это значит, что автомат A допускает w, а значит в A существут путь из стартовой вершины - обозначим ее S, в какую-либо терминальную, причем буквы на пути образуют слово w, и длинна этого пути будет n. Длинна пути - сумма длинн всех ребер на пути, длинна ε-ребра - 0, длинна остальных ребер - 1.

Для решения задачи пострим новый НКА - A' следующим образом:
Если v_i - вершина в A, то в A' создадим вершины $v_i^0, v_i^1, v_i^2 ... v_i^{k-1}$, Так, если в A было n вершин, то в A' будет $n * k$ вершин.

Проведем ребра по следующим правилам:
Если (v_i, v_j, α) - ребро в A из v_i в v_j по символу $\alpha \neq \varepsilon$, то в A' проведем ребра $\forall t \in 0..(k-1)(v_i^t, v_j^{(t+1)\%k}, \alpha)$. Если (v_i, v_j, ε) - ребро в A,

то проведем ребра $\forall t \in 0..(k-1)(v_i^t, v_j^t, \varepsilon)$

Стартовая вершина в A' будет S^0, где S - стартовая вершина в A, если вершина v_i была терминальной в A, то в A' терминальной будет помечена вершина v_i^l

Докажем, что A' задает язык $w \in A' \Leftrightarrow w \in A \wedge |w|\%k = l$, т.е. w принимается автоматом A' тогда и только тогда, когда w принимается A и длинна w по модулю k равна l.

Пусть w принимается A', докажем, что $|w|\%k = l$:
w приниматеся, а значит соответсвует какому-либо пути в A', а так как количество не ε-ребер на этом пути ровно $|w|$ и при каждом проходе по такому ребру верхний индекс вершины увеличивается на 1 по модулю k, то у последний вершины верхний индек в точности $|w|\%k$, а так как слово принимается автоматом, то последняя вершина терминальная, а значит ее верхний индекс l по построению автомата A', а значит $|w|\%k = l$, ЧТД.

Докажем, что если w приниматеся A' то w принимается A:
Рассмотрим путь $v_0^{i_0}, v_1^{i_1}, ...v_n^{i_n}$ соответствующие слову w в A', тогда заметим, что $v_0, v_1, ..v_n$ - путь в A, так как все нужные ребра присутствуют по построению A', и v_n так же терминальная, а значит w приниматеся A, ЧТД

А значит $w \in A' \Rightarrow w \in A \wedge |w|\%k = l$.

Пусть $w \in A \wedge |w|\%k = l$, докажем что $w \in A'$
Пусть $v_0, v_1, ...v_n$ пусть в A, соответствующий w, построим в A' путь, соответствующий w с помощью пути в A:
Начинается путь со стартовой вершины S^0. Пусть мы еще не обработали суффикс $v_i, v_{i+1}, ...v_n$ пути в A, Тогда, пусть мы находимся в вершине v_i^t и обрабатываем ребро (v_i, v_{i+1}, α), причем $\alpha \neq \varepsilon$. Тогда пройдем по ребру $(v_i^t, v_{i+1}^{(t+1)\%k}, \alpha)$, оно есть по построению A'. Если же обрабатываем ребро $(v_i, v_{i+1}, \varepsilon)$, то пройдем по ребру $(v_i^t, v_{i+1}^t, \varepsilon)$, оно так же есть по построению. В любом случае мы всегда можем сделать следующий переход, и всегда сохраняется инвариант, о том, что если мы находимся в вершине v_i^t то обрабатывать мы будем ребро из вершины v_i, а значит мы построим некоторый путь задающий слово w в A', причем, так как мы закончим в вершине, с верхним индексом l(Так как $|w|\%k = l$), то закончим в v_n^l и при этом v_n^l тер-

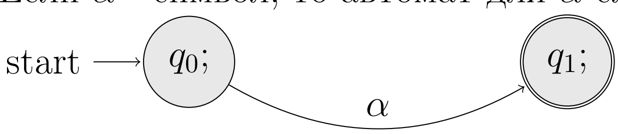
минальная так как v_n терминальная в A, а значит w приниматся A', а значит $w \in A' \Leftarrow w \in A \wedge |w| \% k = l$.
Получим $w \in A' \Leftrightarrow w \in A \wedge |w| \% k = l$, ЧТД

А значит для решения достаточно проверить достижимость конечной вершины в A'

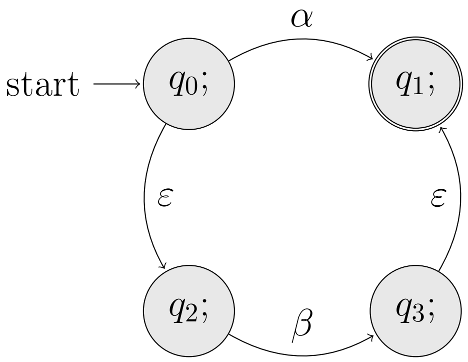
Отметим, что для решения совсем не обязательно строить A' в явном виде, так, если у нас есть A то решение сводится к следующему:
У нас есть Посетитель, который запоминает, в каких вершинах и с каким значением LenMod он был (LenMod - длинна слова, по которому он пришел в вершину по модулю k), и Посетитель обходит граф A, ДФС-ом, проходя по ребру тогда и только тогода, когда проход по ребру приведет его в состояние, в котором он еще не был, не забываем правильно обновлять LenMod при возврате по ребру, и в конце, для получения ответа, проверим, был ли Посетитель в вершине $Terminal$ и состоянием LenMod = l, Сложность решения, очевидно, по времени $O((V + E) * K)$, E - число ребер, V - число вершин, K - это K из условия, по памяти $O(V * K)$.

3 О построении НКА, эквевалентоного регулярному выражению

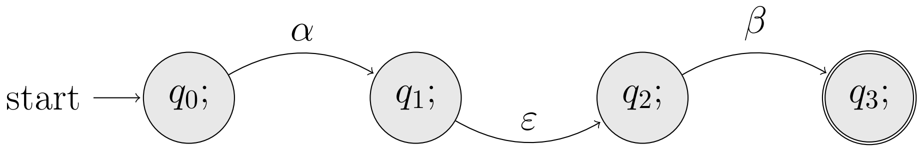
Мы умеем решать задачу если для регулярного выражения у нас есть эквивалентный ему НКА. Покажем, как по регулярному выражению строить НКА.
Для удобства будем пользоваться только НКА с ровно одной терминальной вершиной, при построении будем поддерживать это инвариант. Пусть α, β - регулярные выражения, тогда:
1) Если α - символ, то автомат для α следующий:



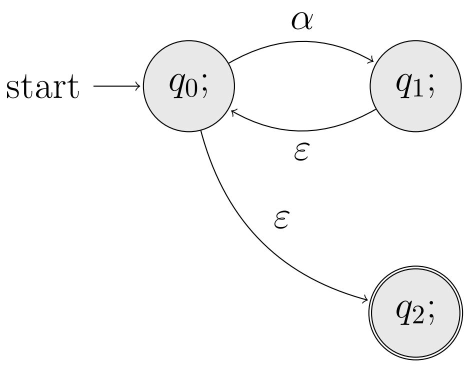
2) Автомат для $\alpha + \beta$:
q_0 и q_1 стартовая и терминальная в автомате для α
q_2 и q_3 стартовая и терминальная в автомате для β



2) Автомат для $\alpha.\beta$:
q_0 и q_1 стартовая и терминальная в автомате для α
q_2 и q_3 стартовая и терминальная в автомате для β



2) Автомат для $\alpha *$:
q_0 и q_1 стартовая и терминальная в автомате для α
q_2 - новая вершина



Оценим сложность построения НКА по регулярному выражению и размер построенного НКА.
В общем случаем, любая операция из $+,.,^*$ а так же операция создания НКА для регулярного выражения из одного символа требует добавления константного числа ребер и константного числа вершин, и, если при операциях можно считать, что операнды можно инвалидировать, то построение будет занимать $O(n)$ времени. Если же инвалидировать операнды нельзя, то построение занимает $O(n^2)$ времени, так как при каждой операции необходимо провести копирование всего НКА, а значит время каждой операции возрастает до

линейного. Кроме того, очевидно по построению, что в полученном НКА $O(n)$ ребер и вершин. n - Длинна регулярного выражения.

4 Итоговая оценка

Таким образом, итоговое время работы $O(n * k)$, так как НКА сторoится за $O(n)$ и в нем $O(n)$ вершин и ребер, а значит поиск ответа займет $O(n * k)$ по времени и $O(n * k)$ по памяти.

5 PS

Существует простое решение данной задачи через динамику, в котором не требуется построение автомата, однако его сложность $O(n*k^2$ по времени и $O(n * k)$ по памяти, хоть и с небольшой константой.