ФЯ ДЗ 14

Задача 1.

Условие: для языка $L = \{(ac)^n(cb)^n | n > 0\}$ построить допускающий его МП-автомат.

Решение:

Давайте взглянем на несколько слов, порождаемых данным языком: accb, acaccbcb, … . Вот такой вот симметричный язык.

Обозначим за начальное состояние q, за конечное f, за символ дна стека E (от английского Empty - пусто). Правила будут следующие:

$\delta(q, ac, E) \to \delta(q, \varepsilon, XE)$

$\delta(q, ac, X) \to \delta(q, \varepsilon, XX)$

$\delta(q, cd, XX) \to \delta(h, \varepsilon, X)$

$\delta(q, cd, XE) \to \delta(f, \varepsilon, E)$ - на случай, если имеется строка accb

$\delta(h, cd, XX) \to \delta(h, \varepsilon, X)$

$\delta(h, cd, XE) \to \delta(f, \varepsilon, E)$

Задача 2.

Условие: для языка $L = \{(ac)^n(ab)^n | n \geq 0\}$ построить допускающий его МП-автомат.

Решение:

В отличие от предыдущего примера, в данном языке также может иметься пустая строка, а в целом язык тот же, если принять “cb” из предыдущего языка за “ab” в этом.

Обозначим за начальное состояние q, за конечное f, за символ дна стека E. Правила:

$\delta(q, \varepsilon, E) \to \delta(f, \varepsilon, E)$ - на случай, если дана пустая строка

$\delta(q, ac, E) \to \delta(q, \varepsilon, XE)$

$\delta(q, ac, X) \to \delta(q, \varepsilon, XX)$

$\delta(q, ab, XX) \to \delta(h, \varepsilon, X)$

$\delta(q, ab, XE) \to \delta(f, \varepsilon, E)$ - на случай, если имеется строка acab

$\delta(h, ab, XX) \to \delta(h, \varepsilon, X)$

$\delta(h, ab, XE) \to \delta(f, \varepsilon, E)$

Задача 3.

Условие: для языка $L = \{10^n1^n | n > 0\} \cup \{110^n1^{2n} | n \geq 0\}$ построить допускающий его МП-автомат.

Решение:

А здесь уже не дурно бы и пораскинуть мозгами…

Для начала назовём слова, порождаемые скобкой $\{10^n1^n | n > 0\}$ - “левой” частью языка. Слова, порождаемые скобкой $\{110^n1^{2n} | n \geq 0\}$ - “правой” частью языка.

Рассмотрим, что такое $\{10^n1^n | n > 0\}$. Это у нас ни что иное, как 101, 10011, 1000111, … . Грубо говоря в начале всегда будет одна единица, далее n нулей и за ними n единиц. Хорошо, это сделать вполне возможно.

Посмотрим теперича, что такое $\{110^n1^{2n} | n \geq 0\}$. Это у нас ни что иное, как 11, 11011, 11001111, 11000111111, … , то есть в начале всегда будет две единички, а далее n нулей и $2n$ единиц.

Задачка нетривиальная…

Поступить можно следующим образом - если в начале входной строки у нас имеется на входе одна 1, а далее 0, это будет значить, что входное слово стоит попытаться отнести к “левой” части языка (то бишь тому, что порождают левые фигурные скобки), и пытаться его распознать, основываясь на свойствах “левой” части. А вот если в начале идёт две единицы, а далее уж пустота или или 0, то стоит целиться на слово “правой” части языка, так как словом из “левой” части языка оно совершенно точно быть не может.

Отработаем механизм “понимания” того, слово из какой части мы распознаём.

Пусть мы считали символ 1. Это ещё ни о чём не говорит. Будем считывать второй символ. Если это слово из “левой” части языка, то мы обязательно встретим 0, так

как в этой части число n обязательно положительное, то бишь хотя бы один 0 и одна 1 нам гарантированы. А вот ежели вторым символом мы встретим 1, то это точно слово из “правой” части языка, так как любое слово из “правой” части языка гарантированно начинается на две единицы.

Таким образом мы будем строить распознаватели для левой и правой части с тем учётом, что первый символ уже считан и мы в данный момент считываем второй. Пусть после считывания первого символа мы переместились из начального состояния q в состояние q_1.

Левая часть:

$\delta(q_1, 0, E) \to \delta(l_1, \varepsilon, XE)$ - из состояния q_1 мы попадаем в состояние l_1 (l от английского left - левый). И сразу же начинаем распознавать симметричную цепочку.

$\delta(l_1, 1, XE) \to \delta(f, \varepsilon, E)$ - на случай, если на вход будет подана строка 101.

$\delta(l_1, 0, X) \to \delta(l_1, \varepsilon, XX)$

$\delta(l_1, 1, XX)\delta(l_2, \varepsilon, X)$

$\delta(l_2, 1, XX) \to \delta(l_2, \varepsilon, X)$

$\delta(l_2, 1, XE) \to \delta(l_2, \varepsilon, E)$

Правая часть:

$\delta(q_1, 1, E) \to \delta(r_0, \varepsilon, E)$ - из состояния q_1 мы считали символ 1, следовательно нам стоит попробовать распознать слово из “правой” части языка.

Стоит учитывать, что “правой” части языка так же принадлежит цепочка 11, значит если в состоянии r_0 мы увидим на входе пустое слово, мы можем перейти в заключительное состояние.

$\delta(r_0, \varepsilon, E) \to \delta(f, \varepsilon, E)$

$\delta(r_0, 0, E) \to \delta(r_0, \varepsilon, XXE)$

$\delta(r_0, 1, XX) \to \delta(r_1, \varepsilon, X)$

$\delta(r_1, 1, XX) \to \delta(r_1, \varepsilon, X)$

$\delta(r_1, 1, XE) \to \delta(f, \varepsilon, E)$

Объединим:

$\delta(q_1, 0, E) \to \delta(l_1, \varepsilon, XE)$

$\delta(q_1, 1, E) \to \delta(r_0, \varepsilon, E)$

$\delta(l_1, 1, XE) \to \delta(f, \varepsilon, E)$

$\delta(l_1, 0, X) \to \delta(l_1, \varepsilon, XX)$

$\delta(l_1, 1, XX)\delta(l_2, \varepsilon, X)$

$\delta(l_2, 1, XX) \to \delta(l_2, \varepsilon, X)$

$\delta(l_2, 1, XE) \to \delta(l_2, \varepsilon, E)$

$\delta(r_0, \varepsilon, E) \to \delta(f, \varepsilon, E)$

$\delta(r_0, 0, E) \to \delta(r_0, \varepsilon, XXE)$

$\delta(r_0, 1, XX) \to \delta(r_1, \varepsilon, X)$

$\delta(r_1, 1, XX) \to \delta(r_1, \varepsilon, X)$

$\delta(r_1, 1, XE) \to \delta(f, \varepsilon, E)$

Задача 4.

Для языка $L = \{a^n b^m c^k | n + k \neq m\}$ построить допускающий его МП-автомат.

То есть всего лишь навсего требуется, чтобы суммарное количество символов a и символов c не равнялось количеству символов b. Тогда поступим таким образом. Пока будем считывать символы a, будем класть в стек символ A. Далее, считывая символы b, будем класть в стек символы B. После этого, считывая символы c, будем класть в стек символы C. Когда строка закончилась, начнём обрабатывать стек. Теперь поступим следующим образом. Представим, что у нас есть какое-то количество состояний автомата, пусть количество состояний это c, а состояния будем обозначать как s_i.

Идея состоит в следующем. Пусть мы прочитали какую-то строчку $a^n b^m c^k$. В стеке теперь у нас находится: $\underset{\text{k штук}}{\underbrace{C...C}}$ $\underset{\text{m штук}}{\underbrace{B}}$ $\underset{\text{n штук}}{\underbrace{A}}$ E. Начнём извлекать из стека символы, и вместе с этим двигаться по состояниям автомата. После того, как считана цепочка, мы находимся в состоянии s_0 (во время считывания цепочки мы не предпринимали никаких действий). И действовать будем следующим образом:

Если мы находимся в состоянии s_i, то:

- Если мы извлекли из стека C, то переместимся в состояние s_{i+1}
- Если мы извлекли из стека B, то переместимся в состояние s_{i-1}

- Если мы извлекли из стека A, то переместимся в состояние s_{i+1}

Таким образом в том случае, если условие $n + k \neq m$ не выполняется и всё-таки $n + k = m$, то автомат по окончании работы окажется в состоянии s_0 (то бишь в начальном). Это будет значить, что цепочка языку не принадлежит. Во всех же остальных случаях цепочка языку принадлежать будет. Стоит отметить, что у нас могут быть состояния с отрицательными индексами (пример: $n = 1, m = 3, c = 2$ - состояния можно разложить как s_{-2}, s_{-1}, s_0, s_1). Мы не знаем, сколько их будет, однако мы знаем, что их счётное множество, прямо как множество слов, порождаемых языком L.

Собственно, можно перемещаться по состояниям s_i прямо в тот момент, когда мы считываем очередной символ, поэтому разбиение этого процесса на две части (сначала заполнение стека, а потом перемещение по состояниям) особо и не имеет смысла. Однако в таком случае мы можем обойтись и без заполнения стека. В таком случае будем использовать стек, чтобы понять, достигли ли мы конечного состояния. В таком случае пусть по завершении получения слова, если мы оказались вдруг в s_0, положим в стек символ X - это будет признаком непринятия цепочки. В противном случае, если мы по завершении цепочки оказались не в состоянии s_0, в стек положим Y. Потом будем двигаться из состояния s_i к состоянию s_0, а в самом s_0 проверим, если в стеке имеется символ X - не делать ничего, в противном случае принять цепочку.

$\delta(s_i, a, E) \to \delta(s_{i+1}, \varepsilon, E)$

$\delta(s_i, b, E) \to \delta(s_{i-1}, \varepsilon, E)$

$\delta(s_i, c, E) \to \delta(s_{i+1}, \varepsilon, E)$

$\delta(s_0, \varepsilon, E) \to \delta(s_0, \varepsilon, XE)$ - мы достигли конца цепочки и оказались в s_0

$\forall i > 0 : \delta(s_i, \varepsilon, E) \to \delta(s_{i-1}, \varepsilon, YE)$ - мы достигли конца цепочки, начинаем двигаться к s_0

$\forall i < 0 : \delta(s_i, \varepsilon, E) \to \delta(s_{i+1}, \varepsilon, YE)$ - мы достигли конца цепочки, начинаем двигаться к s_0

$\delta(s_0, \varepsilon, YE) \to (f, \varepsilon, E)$, f - заключительное состояние.