

Домашнее задание 23

Шумилкин Андрей, группа 163

Задача 1

Как нам известно из тезиса Чёрна-Тьюринга всякая вычислимая функция вычислима машиной Тьюринга, а нигде не определенная функция является вычислимой, поскольку достаточно просто "зациклить" программу, то есть чтобы она уклонилась от выдачи какого-либо ответа – это и будет алгоритм ее вычисления.

Значит существует МТ, которая вычисляет нигде не определенную функцию.

Пусть A – входной алфавит и q – некоторый символ из $A \cup \{\Lambda\}$ Тогда нам достаточно просто зациклить МТ и мы можем сделать это следующей таблице переходов:

$$\delta := \left\{ (q, 0) \mapsto (q(\text{тот же самый}), 0, +1) \right\}$$

Где 0 – начальное состояние.

Действительно, согласно таблице, машина сначала просто "пройдет" по входным данным в виде числа, а потом продолжит двигаться вправо по пустым символам бесконечно, так как лента машины бесконечна.

Задача 2

Приведем таблицу переходов:

$$\delta := \left\{ \begin{array}{l} (0, 0) \mapsto (1, 0, +1) \\ (1, 0) \mapsto (0, 0, +1) \\ (\Lambda, 0) \mapsto (\Lambda, 0, 0) \end{array} \right.$$

Где 0 – начальное состояние.

Заметим, что если головка находится над символом из алфавита w , то есть над 0 или 1, то она сдвинется вправо, предварительно записав в предыдущую ячейку "отрицание" текущего символа, то есть инвертирует его.

Когда же головка дойдет до пробельного символа она просто остановится, при этом к этому моменту все символы входа из w будут инвертированы, что по определению, данному в задаче, и будет представлять инвертированное входное слово.

Задача 3

Пусть q – любая буква входного алфавита, а l – любой символ из символов входного алфавита и Λ .

Приведем таблицу переходов:

$$\delta := \left\{ \begin{array}{l} (a, 0) \mapsto (a, 1, +1) \\ (b, 1) \mapsto (b, 2, +1) \\ (a, 2) \mapsto (a, 3, +1) \\ (q, 3) \mapsto (q, 3, +1) \\ (\Lambda, 3) \mapsto (\Lambda, 4, -1) \\ (q, 4) \mapsto (\Lambda, 4, -1) \\ (\Lambda, 4) \mapsto (1, 5, +1) \\ (a, 1) \mapsto (a, 0, +1) \\ (b, d \in \{0, 2\}) \mapsto (b, 0, +1) \\ (c, d \in \{0, 1, 2\}) \mapsto (c, 0, +1) \\ (\Lambda, d \in \{0, 1, 2\}) \mapsto (\Lambda, 6, -1) \\ (q, 6) \mapsto (\Lambda, 6, -1) \\ (\Lambda, 6) \mapsto (0, 5, +1) \\ (\Lambda, 5) \mapsto (\Lambda, 5, 0) \end{array} \right.$$

Где 0 – начальное состояние.

Действительно, если головка наткнется на последовательность *aba*, то машина перейдет в состояние 3, после чего просто сдвинется до ближайшего справа Λ , перейдет в состояние 4 и пойдет влево, до ближайшего Λ , попутно «затирая» все входные данные. Когда она дойдет до Λ , то у нас на ленте будут только пробельные символы, а значит достаточно лишь вывести ответ -1 , поскольку мы в состоянии 4, а в него можно попасть, только если мы нашли *aba*. Поэтому мы ставим один и переходим в состояние 5, а также сдвигаемся на одну клетку вправо, где стоит Λ и, согласно таблице переходов, останавливаемся. Таким образом в случае нахождения последовательности символов *aba* машина работает правильно.

Если же машина не находит *aba*, то она так и идет до ближайшего справа Λ , не доходя до состояния 3 и когда она приходит в него она переходит в состояние 6 и идет до ближайшего слева Λ , попутно «затирая» входные данные, подобному случаю, когда мы все-таки нашли *aba*, только в данном случае у нее другое состояние -6 . Придя в ближайшее слева Λ она выводит ответ -0 , и переходит в состояние 5, сдвигается вправо и останавливается. Таким образом в случае ненахождения последовательности символов *aba* машина также работает правильно.

Задача 4

Опишем общую идею алгоритма, который будем реализовывать с помощью МТ.

У нас сначала должны идти нули, а потом единицы, то есть по итогу у нас не долж-

но быть подслов вида 10, но кол-во 0 и 1 не должно измениться. Значит нам нужно преобразовывать 10 в 01.

Тогда мы можем просто ходить по слову и искать самое левое 10, преобразовывать его в 01 и возвращаться в начало слова, и так до тех пор, пока 10 не останется, то есть пока мы не дойдем до ближайшего справа Λ , не найдя ни одного.

Мы можем реализовать следующим образом. Пусть s_0 – начальное состояние. Начинаем двигаться влево, сохраняя его и символы, если встречаем нули. Если же мы встречаем 1, то переходим в s_1 и сдвигаемся вправо далее, если далее встречаем 0, находясь в состоянии s_1 , то переходим в s_2 , пишем на его месте 1, сдвигаемся влево. Далее пишем 0, снова сдвигаемся влево и переходим в состояние s_3 .

В состоянии s_3 мы должны просто двигаться влево до ближайшего *Lambda* и, дойдя до него, не изменяя его, сдвинуться вправо и перейти в состояние s_0 для которого уже описаны действия выше.

Если же мы встречаем Λ в состоянии s_1 или s_0 , то мы просто останавливаемся, потому что это значит, что мы не встретили по пути 10 и слово уже отсортировано в требуемом нам порядке.

Поскольку мы смогли описать алгоритм в виде для МТ мы и можем построить такую МТ, то есть она существует.