Домашняя работа по дискретной математике №23

Михайлов Никита Маратович, ПМИ-167.

Комментарий. В решении таблицы описаны следующим образом:

$$A : b \to c : \{R, L, N\} : D,$$

что означает «в состоянии A символ b переходит в символ c, после чего двигается вправо(R), влево(L) или стоит на месте(N) и переходит в состояние D». А так же будем считать, что λ – пробельный символ и * – любой символ из данного алфавита.

Задание 1.

Решение. Если алгоритм, вычисляющий некоторую функцию зацикливается на любом входе, то эта функция нигде не определена.
Пусть начальное состояние q1. Опишем таблицу переходов:

$$q1 : * \to \lambda : R : q1$$

Тогда алгоритм просто будет печатать пробельный символ λ и никогда не остановится.

Задание 2.

Решение. Объясним на словах. В состоянии q1 инвертивуем символ, передвигаемся вправо и остаемся в этом же состоянии. Когда встречаем пробельный символ идем влево и переходим в состояние q2, в котором возвращаемся обратно. Опишем таблицу переходов:

$$q1 : 1 \to 0 : R : q1$$
$$q1 : 0 \to 1 : R : q1$$
$$q1 : \lambda \to \lambda : L : q2$$
$$q3 : 1 \to 1 : L : q2$$
$$q3 : 0 \to 0 : L : q2$$

$\boxed{\text{Задание 3.}}$

Решение. Снова опишем сначала словами.Пусть начальное состояние q. В состоянии q ищем вхождение символа a. Если нашли, то переходим в состояние $FoundA$, если не нашли, то печатаем 0 и переходим в состояние $notFoundABA$ (которое удаляет все, и печатает в конце 0). В состоянии $FoundA$ ищем сразу после символ b. Если не нашли b, то переходим в начальное состояние q. Если нашли b, то переходим в состояние $foundB$ и ищем a. И снова: если не нашли, то переходим в состояние q, а если нашли, то переходим в состояние $runnerToEnd$(которое бежит до конца, а когда видит пробельный символ останавливается и переходит в состояние $foundABA$, который все стирает и печатает 1). Прилагается Ссылка. на код. Вставьте этот код сюда.

Опишем таблицу состояний:

$$
\begin{aligned}
&q : a \to a : R : foundA\\
&q : b \to b : R : q\\
&q : c \to c : R : q\\
&q : \lambda \to \lambda : L : notFoundABA\\
&foundA : b \to b : R : foundB\\
&foundA : a \to a : N : q\\
&foundA : c \to c : N : q\\
&foundA : \lambda \to \lambda : N : q\\
&foundB : a \to a : R : runnerToEnd\\
&foundB : b \to b : N : q\\
&foundB : c \to c : N : q\\
&foundB : \lambda \to \lambda : N : q\\
&notFoundABA : a \to \lambda : L : notFoundABA\\
&notFoundABA : b \to \lambda : L : notFoundABA\\
&notFoundABA : c \to \lambda : L : notFoundABA\\
&notFoundABA : \lambda \to 0 : N : notFoundABA\\
&FoundABA : a \to \lambda : L : FoundABA\\
&FoundABA : b \to \lambda : L : FoundABA\\
&FoundABA : c \to \lambda : L : FoundABA\\
&FoundABA : \lambda \to 1 : N : foundABA\\
&runnerToEnd : a \to a : R : runnerToEnd\\
&runnerToEnd : b \to b : R : runnerToEnd\\
&runnerToEnd : c \to c : R : runnerToEnd\\
&runnerToEnd : \lambda \to \lambda : L : FoundABA
\end{aligned}
$$

Задание 4.

Решение. Будем использовать трехленточную МТ. На вход поступает двоичное слово на первую ленту. Запишем на вторую ленту a нулей, а на третью b единиц, используя счетчик(счетчик до n реализуется через n состояний). Теперь удалим все с первой ленты и скопируем туда сначала все со второй, а потом все с третьей лент. Копирование происходит по принципу: $(*, 1, *) \to (1, 1, *)$.

Задание 5.

Решение. Скопируем слово на вторую ленту. Передвинем каретку на второй ленте в конец слова. И одновременно будем идти по двум лентам, сравнивая буквы. Как сравнивать? Если (a, a), то продолжаем работу в данном состоянии, а если (a, b), то переходим в состояние, после которого мы все удаляем и выводим 0. Состояние проверки заканчивается когда мы встречаем (λ, λ).

Задание 6.

Решение. Будем использовать 2018 лент. В состоянии 1 $(*, *, ..., *) \to (\lambda, 0, 0, ..., 0)$ – на первой ленте ничего не меняем, а на остальных 2017-ти лентах запишем 0. Далее в состоянии 2 опишем таблицу:
$\forall i \in [1, 2017]$ сделаем переход$(\lambda_0, 1_1, 1_2, ..., 0_i, ...0_{2017}) \to (1_0, 1_1, 1_2, ..., 1_i, ..., 0_{2017})$ и переводим головку на первой ленте вправо. Почему это работает? Для $i = 1$ у нас на первой ленте только пробельные символы, а на всех остальных нули. Данная конфигурация приведет к тому, что на первую ленту мы запишем единицу, а на вторую поставим 1, тем самым "пометив"ее. Для каждого i у нас выполняется свойство: в конфигурации на местах с индексами от 1 до i стоят единицы. Для конфигурации $(\lambda_0, 1_1, 1_2, ..., 1_{2017})$ перейдем в конечное состояние, которое ничего не делает, тем самым закончив работу нашей МТ. Будет записано 2017 единиц.

Задание 7.

Решение. Заметим, что на семинаре была доказана биекция $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N} : f(x, y) = \binom{x + y + 1}{2} + x = \dfrac{(x + y)(x + y + 1)}{2} + x$. Числа подаются на вход в унарном виде. Складывать мы умеем на МТ($x + y$: к 1^x приписываем y единиц), умножать тоже умеем(xy : x раз пишем 1^y). Остается проблема с делением на 2. Нетрудно заметить, что умножатся числа разной четности, поэтому числитель дроби делится на 2 без остатка. Пусть вычислили числитель. Он состоит из единиц. Будем выписывать единицу на другую ленту если находимся в первом состоянии, после чего переходим во второе и просто пропускаем единицу, переходя снова в первое состояние. Закончим работу, когда встретим разделитель. На выходе сложим выписанное число с числом x.