Дополнительные задачи по МТ

1. Запрещенные вхождения

Написать программу МТ, которая вычисляет такую функцию:

$$f(x)=\begin{cases}1, \text{ если } abaa \prec x, \text{ но } bab\neg \prec x\\ 0 \text{ иначе}\end{cases}$$

Идея решения: при проходе по ленте слева направо искать запрещенное вхождение, и если оно найдется, то вернуться и выдать 0. В противном случае искать нужное вхождение при движении по ленте справа налево, одновременно стирая слово, то есть искать вхождение инверсии нужного слова.

Программа:

$q_0 * \to q_0 *, R$
$q_0 a \to q_0 a, R$
$q_0 b \to q_1 b, R$
$q_1 a \to q_2 a, R \,//\, ba$
$q_1 b \to q_1 b, R \,//\, bb$
$q_2 a \to q_0 a, R \,//\, baa$
$q_2 b \to q_3 b, R \,//\, bab \prec x$

Переход в состояние q_3 означает, что найдено запрещенное вхождение. Далее следует дочитать входное слово до конца, при возврате стереть его и выдать 0 как результат.

$q_3 \alpha \to q_3 \alpha, R \,//\, \alpha \in \{a,b\}$
$q_3 \square \to q_4 \square, L$
$q_4 \alpha \to q_4 \square, L$
$q_4 * \to q_4 *, R$
$q_4 \square \to q_f 0, L$

Следующий фрагмент программы отрабатывает процесс поиска нужного вхождения при условии, что запрещенное вхождение не найдено:

$q_i \square \to r_0 \square, L \,//\, i = 0,1,2$
$r_0 * \to r_0 *, R$
$r_0 \square \to q_f 0, L$

Если запрещенное вхождение не найдено, то начинается возврат (движение головки справа налево) в состоянии r_0. Если в этом состоянии после шага влево сразу виден маркер начала ленты, то это означает, что входное слово пустое, которое, естественно, распознается как «неправильное».

Далее ищем инверсию нужного вхождения:

$r_0 a \to r_1 \square, L$

$r_0 b \to r_0 \square, L$

$r_1 a \to r_2 \square, L$ / / найден префикс aa инверсии

$r_1 b \to r_0 \square, L$ / / $\overleftarrow{ba}$

$r_2 a \to r_2 \square, L$ / / ... $\overleftarrow{aaa}$

$r_2 b \to r_3 \square, L$ / / $\overleftarrow{baa}$

$r_3 a \to r_4 \square, L$ / / $\overleftarrow{abaa}$ (найдено нужное вхождение)

$r_3 b \to r_0 \square, L$ / / $\overleftarrow{bbaa}$

$r_4 \alpha \to r_4 \square, L$ / / $\alpha \in \{a, b\}$

$r_4 * \to r_4 *, R$

$r_4 \square \to q_f 1, L$ / / $success!$

$r_i * \to r_i *, R$ / / $i = 1, 2, 3;$ нужное вхождение не найдено

(команда с левой частью r_0 * уже есть)

$r_i \square \to q_f 0, R$ / / $i = 1, 2, 3$

Последняя команда выполняется в случае, если не найдено ни одно вхождение: ни запрещенное, ни нужное. Это покрывает и случай пустого слова на входе. Команды с левой частью r_0 * и затем с левой частью $r_0 \square$ (см. фрагмент выше) выполняются всякий раз, когда маркер обозревается в состоянии r_0. Это происходит, в частности, и при пустом входном слове.

2. Нормальный алгорифм (НА), решающий ту же задачу:

$$\begin{cases} 0\xi \to 0 \,/\,/\, \xi \in \{a, b\} \\ \xi 0 \to 0 \\ 0 \to \cdot 0 \\ bab \to 0 \\ 1\xi \to 1 \\ \xi 1 \to 1 \\ 1 \to \cdot 1 \\ abaa \to 1 \\ \xi \to 0 \\ \to \cdot 0 \end{cases}$$

Последняя формула применяется только к пустому входному слову, а вторая снизу – к входному слову, не содержащему вхождений ни одного из двух слов.

3. МТ, выполняющая обращение входного слова.
   Записанная ниже программа копирует входное слово $x \in V^*$ в обратном порядке, замещая его на ленте решетками. После этого надо передать управление программе сдвига вправо (см. файл «Примеры машин Тьюринга»). Это можно сделать и в самой программе инверсного копирования. Вводится, как обычно, алфавит букв-двойников.

$$q_0 * \to q_0 *, R$$
$$q_0 \square \to q_f \square, L \,//\, x = \lambda$$
$$q_0 \alpha \to q_1 \alpha, R \,//\, \alpha \in V$$
$$q_1 \alpha \to q_1 \alpha, R$$
$$q_1 \square \to q_2 \square, R$$
$$q_2 \alpha \to q_\alpha \#, R$$
$$q_\alpha \square \to q_2 \overline{\alpha}, L$$
$$q_\alpha \overline{\beta} \to q_\alpha \overline{\beta}, R \,//\, \alpha, \beta \in V$$
$$q_2 \overline{\alpha} \to q_2 \overline{\alpha}, L$$
$$q_2 \# \to q_2 \#, L$$
$$q_\alpha \# \to q_\alpha \#, R$$
$$q_2 * \to q_3 *, R$$
$$q_3 \# \to q_3 \#, R$$
$$q_3 \overline{\alpha} \to q_3 \alpha, R$$
$$q_3 \square \to q_4 \square, L$$

Следующий фрагмент программы выполняет сдвиг инверсии слова влево на число ячеек, равное длине слова:

$$q_4 \alpha \to r_\alpha \square, L$$
$$r_\alpha \beta \to r_\beta \alpha, L$$
$$r_\alpha \# \to r_\# \alpha, L$$
$$r_\# \# \to q_3 \#, R$$
$$q_3 \alpha \to q_3 \alpha, R$$
$$r_\# * \to q_f *, S$$

Прогонка:

$(q_0, \lambda, *abc\square) \mapsto (q_0, *, abc\square) \mapsto (q_1, *a, bc\square) \mapsto^2 (q_1, *abc, \square) \mapsto$
$\mapsto (q_2, *ab, c\square) \mapsto (q_c, *ab\#, \square) \mapsto (q_2, *ab, \#\overline{c}\square) \mapsto (q_2, *a, b\#\overline{c}\square) \mapsto$
$\mapsto (q_b, *a\#, \#\overline{c}, \square) \mapsto (q_b, *a\#\#, \overline{c}\square) \mapsto (q_b, *a\#\#\overline{c}, \square) \mapsto (q_2, *a\#\#, \overline{c}\overline{b}\square) \mapsto^2$
$\mapsto^2 (q_2, *, a\#\#\overline{c}\overline{b}\square) \mapsto (q_a, *\#, \#\#\overline{c}\overline{b}\square) \mapsto^4 (q_a, *\#\#\#\overline{c}\overline{b}, \square) \mapsto (q_2, *\#\#\#\overline{c}, \overline{b}\overline{a}\square)$
$\mapsto^5 (q_2, *, \#\#\#\overline{c}\overline{b}\overline{a}\square) \mapsto (q_2, \lambda, *\#\#\#\overline{c}\overline{b}\overline{a}\square)$

Получена копия (в буквах-двойниках) инверсии, сдвинутая на 3 ячейки вправо. Далее переходим в исходный алфавит и сдвигаем слово влево на 3 ячейки:

$(q_2, \lambda, *\#\#\#\overline{c}\overline{b}\overline{a}\square) \mapsto (q_3, *, \#\#\#\overline{c}\overline{b}\overline{a}\square) \mapsto^3 (q_3, *\#\#\#, \overline{c}\overline{b}\overline{a}\square) \mapsto (q_3, *\#\#\#c, \overline{b}\overline{a}\square) \mapsto^2$
$\mapsto^2 (q_3, *\#\#\#cba, \square) \mapsto (q_4, *\#\#\#cb, a\square) \mapsto (r_a, *\#\#\#c, b\square\square) \mapsto (r_b, *\#\#\#, ca\square)$
$\mapsto (r_c, *\#\#, \#ba\square) \mapsto (r_\#, *\#, \#cba\square) \mapsto (q_3, *\#\#, cba\square)...$

и т. д.

4. Построить МТ, вычисляющую следующую функцию:

$$f(x) = \begin{cases} 1, \text{ если слово } aab \text{ входит в слово } x \in \{a,b,c\}^* \text{ и слово} \\ cab \text{ входит в } x \\ 0 \text{ иначе} \end{cases}$$

(ср. с задачей №4 из файла «Примеры машин Тьюринга»: здесь надо найти вхождение *каждого* из слов.)

$q_0 * \to q_0 *, R$
$q_0 a \to q_1^a a, R$ / /возможно, это 1-я буква 1-го слова
$q_0 b \to q_0 b, R$
$q_0 c \to q_1^c c, R$ / /возможно, это 1-я буква 2-го слова
$q_1^a a \to q_2^a a, R$ / /прочитано aa
$q_1^a b \to q_0 b, R$ / /прочитано ab, возврат к началу поиска
$q_1^a c \to q_1^c c, R$ / /прочитано ac; буква c может быть 1-й буквой 2-го слова
$q_2^a a \to q_2^a a, R$ / /ждем 3-ю букву 1-го слова
$q_2^a b \to q_3^a b, R$ / / найдено 1-е слово
$q_2^a c \to q_1^c c, R$ / / прочитано aac
$q_1^c a \to q_2^c a, R$ / /прочитано ca
$q_1^c b \to q_0 b, R$ / /прочитано cb, возврат к началу поиска
$q_1^c c \to q_1^c c, R$ / /ждем 2-ю букву 2-го слова
$q_2^c a \to q_2^a a, R$ / /прочитано caa (если следующая буква окажется b, то будет найдено 1-е слово)
$q_2^c b \to q_3^c b, R$ / /найдено 2-е слово
$q_2^c c \to q_1^c c, R$ / /прочитано cac

Переход в состояние q_3^a (q_3^c) означает начало поиска слова cab после того как найдено слово aab (vice versa). Заметим, что вхождения этих слов не могут пересекаться, и при чтении ленты слева направо какое-то из них будет найдено первым (если, конечно, хотя бы одно из них входит в исходное слово).
Продолжаем писать программу:

$q_3^a a \to q_3^a a, R$

$q_3^a b \to q_3^a b, R$

$q_3^a c \to r_1 c, R$

$r_1 a \to r_2 a, R \,//\, ca$

$r_1 b \to q_3^a b, R \,//\, cb$

$r_1 c \to r_1 c, R \,//\, cc$

$r_2 a \to q_3^a a, R \,//\, caa$

$r_2 b \to r_3 b, R \,//\, cab$ – найдено вхождение

$r_2 c \to r_1 c, R \,//\, cac$

Это был фрагмент программы, описывающий поиск слова cab после того как найдено слово aab. Следующий (последний) фрагмент описывает поиск слова aab после того как найдено слово cab:

$q_3^c a \to p_1 a, R$

$q_3^c b \to q_3^c b, R$

$q_3^c c \to q_3^c c, R$

$p_1 a \to p_2 a, R \,//\, aa$

$p_1 b \to q_3^c b, R \,//\, ab$

$p_1 c \to q_3^c c, R \,//\, ac$

$p_2 a \to p_2 a, R \,//\, aaa$

$p_2 b \to r_3 b, R \,//\, aab$ – найдено вхождение

$p_2 c \to q_3^c c, R \,//\, aac$

Команды завершения работы после успешного поиска (то есть после перехода в состояние r_3):

$r_3\alpha \to r_3\alpha, R \,//\, \alpha \in \{a,b,c\}$

$r_3\square \to q_4\square, L$

$q_4\alpha \to q_4\square, L$

$q_4 * \to q_4 *, R$

$q_4\square \to q_f 1, L$

Обработка неуспешного поиска:

$s\square \to q_5\square, L \,//\, s \in \{q_0, q_1^a, q_2^a, q_3^a, q_1^c, q_2^c, q_3^c, r_1, r_2, p_1, p_2\}$

$q_5\alpha \to q_5\square, L \,//\, \alpha \in \{a,b,c\}$

$q_5 * \to q_5 *, R$

$q_5\square \to q_f 0, L$

5. Построить МТ, вычисляющую следующую функцию:

$$f(x) = \begin{cases} 1, \text{ если слова } aba \text{ и } bab \text{ входят в слово } x \in \{a,b\}^* \\ 0 \text{ иначе} \end{cases}$$

Надо найти либо вхождение $abab$, либо вхождение $baba$ (слова наименьшей длины содержащие вхождения обоих слов); либо эти непересекающиеся вхождения в любой последовательности.

$q_0 * \to q_0 *, R$

$q_0\square \to q_f 0, L \,//\, x = \lambda$

$q_0 a \to q_1^a a, R$

$q_0 b \to q_1^b b, R$

$q_1^a a \to q_1^a a, R \,//\, aa$

$q_1^a b \to q_2^a b, R \,//\, ab$

$q_1^b a \to q_2^b a, R \,//\, ba$

$q_1^b b \to q_1^b b, R \,//\, bb$

$q_2^a a \to q_3^a a, R \,//\, aba!$

$q_2^a b \to q_1^b b, R \,//\, abb$

$q_3^a a \to q_1 a, R \,//\, abaa$ (после aba ищем bab)

$q_3^a b \to q_2 b, R \,//\, abab!$

$q_2^b a \to q_1^a a, R \,//\, baa$

$q_2^b b \to q_3^b b, R \,//\, bab!$

$q_3^b a \to q_2 a, R \,//\, baba!$

$q_3^b b \to q_3 b, R \,//\, babb$(после bab ищем aba)

Переход в состояние q_2 означает конец поиска: найдено вхождение $abab$, либо вхождение $baba$.

Следующий фрагмент программы описывает поиск вхождения bab после aba или наоборот, а также завершение работы.

$q_1a \to q_1a, R$ / /продолжаем искать bab после aba

$q_1b \to q_4b, R$ / /$(a)b$

$q_4a \to q_5a, R$ / /$(a)ba$

$q_4b \to q_4b, R$ / /$(a)bb$

$q_5a \to q_1a, R$ / /$(a)baa$

$q_5b \to q_2b, R$ / /$(a)bab!$

$q_3a \to q_6a, R$ / /$(b)a$ продолжаем искать aba после bab

$q_3b \to q_3b, R$ / /$(b)b$

$q_6a \to q_6a, R$ / /$(b)aa$

$q_6b \to q_7b, R$ / /$(b)ab$

$q_7a \to q_2a, R$ / /$(b)aba!$

$q_7b \to q_3b, R$ / /$(b)abb$

$q_2\alpha \to q_2\alpha, R$ / /$\alpha \in \{a,b\}$

$q_2\square \to q_8\square, L$ / /$success!$

$q_8\alpha \to q_8\square, L$

$q_8* \to q_8*, R$

$q_8\square \to q_f 1, L$

$r\square \to q_9\square, L$ / /$r \neq q_0, q_2, q_8, q_9, unsuccess$

$q_9\alpha \to q_9\square, L$

$q_9* \to q_9*, R$

$q_9\square \to q_f 0, L$

Примеры работы

1) $(q_0, \lambda, *\square) \mapsto (q_0, *, \square) \mapsto (q_f, \lambda, *\square)$
2) $(q_0, \lambda, *aabaa\square) \mapsto (q_0, *, aabaa\square) \mapsto (q_1^a, *a, abaa\square) \mapsto (q_1^a, *aa, baa\square) \mapsto$
$\mapsto (q_2^a, *aab, aa\square) \mapsto (q_3^a, *aaba, a\square) \mapsto (q_1, *aabaa, \square) \mapsto$
$\mapsto (q_9, *aaba, a\square) \mapsto^5 (q_9, \lambda, *\square) \mapsto (q_9, *, \square) \mapsto (q_f, \lambda, *0)$

3)

$$(q_0,\lambda,*abab\square)\mapsto(q_0,*,abab\square)\mapsto(q_1^a,*a,bab\square)\mapsto(q_2^a,*ab,ab\square)\mapsto$$
$$\mapsto(q_3^a,*aba,b\square)\mapsto(q_2,*abab,\square)\mapsto(q_8,*aba,b\square)\mapsto^3(q_8,\lambda,*\square)\mapsto$$
$$\mapsto(q_8,*,\square)\mapsto(q_f,\lambda,*1)$$

4)

$$(q_0,\lambda,*bbabbabaa\square)\mapsto(q_0,*,bbabbabaa\square)\mapsto(q_1^b,*b,babbabaa\square\square)\mapsto$$
$$\mapsto(q_1^b,*bb,abbabaa\square)\mapsto(q_2^b,*bba,bbabaa\square)\mapsto(q_3^b,*bbab,babaa\square)\mapsto$$
$$\mapsto(q_3,*bbabb,abaa\square)\mapsto(q_6,*bbabba,baa\square)\mapsto(q_7,*bbabbab,aa\square)\mapsto$$
$$\mapsto(q_2,*bbabbaba,a\square)\mapsto(q_2,*bbabbabaa,\square)\mapsto(q_8,*bbabbaba,a\square)\mapsto^9$$
$$\mapsto^9(q_8,\lambda,*\square)\mapsto(q_8,*,\square)\mapsto(q_f,\lambda,*1)$$

6. Модификация программы сдвига справа налево.
Цель – устранить недочет, связанный с обработкой входного слова без решеток (там возникает тупиковая конфигурация, так как нет команды с левой частью $q_\alpha *(\alpha\in V)$)
Нужно поменять программу так, чтобы входное слово без решеток оставалось без изменений (в частности, пустое входное слово).
Написанная ниже программа учитывает это.

$q_0*\to q_0*,R$
$q_0\alpha\to q_f\alpha,L//\alpha\in V$ (нет решеток)
$q_0\square\to q_f\square,L//$пустое слово на входе (без решеток)
$q_0\#\to q_1\#,R$
$q_1\beta\to q_1\beta,R//\beta\in V\cup\{\#\}$
$q_1\square\to q_2\square,L$
$q_2\#\to q_2\square,L//$пустое слово на входе после решеток; просто стираем все "решетки"
$q_2*\to q_f*,S//$и заканчиваем
$q_2\alpha\to q_\alpha\square,L//$последняя буква сдвигаемого слова стирается
и запоминается в состоянии; $\alpha\in V$
$q_\alpha\beta\to q_\beta\alpha,L//$команды "обмена"; $\alpha,\beta\in V$
$q_\alpha\#\to q_\#\alpha,L//$ очередная стираемая "решетка" заменяется 1-й буквой
сдвигаемого слова;$\alpha\in V$
$q_\#\#\to q_1\#,R//$если стертая "решетка" не последняя, процесс повторяется
$q_\#*\to q_f*,S//$стертая "решетка" последняя, процесс завершается