Некоторые задачи по нормальным алгорифмам (НА)

1. Написать схему НА, аннулирующего всякое непустое слово в заданном произвольном алфавите V, содержащего ровно два вхождения заданного произвольного непустого слова $u = u(1)u(2)...u(k), k \geq 1$. При этом, если входное слово пустое (λ), то результатом должен быть символ @, а если входное слово непусто и не удовлетворяет условию, то результатом должно быть оно само (т. е. алгорифм должен вычислять тождественную функцию).

Схема:

$$\begin{cases} \#\#u \to \cdot u & (1) \\ \#\#\xi \to \xi\#\#/ / \xi \in V & (2) \\ \#\# \to \$ / / \$ \notin V & (3) \\ \xi\$ \to \$ & (4) \\ \$ \to \cdot & (5) \\ \#u \to u(1)\#\#u(2)...u(k) & (6) \\ \#\xi \to \xi\# & (7) \\ \# \to \cdot & (8) \\ u \to u(1)\#u(2)...u(k) / / \# \notin V & (9) \\ \xi \to \cdot\xi & (10) \\ \to \cdot @ / / @ \notin V & (11) \end{cases}$$

Во всей схеме ξ - параметр, обозначающий произвольную букву алфавита V.

Если входное слово пустое, то первой подходящей формулой будет формула (11), «печатается» «собака», и процесс заканчивается. Если входное слово не пустое, и не содержит ни одного вхождения слова $u = u(1)u(2)...u(k), k \geq 1$, то сразу применяется формула, соответствующая строке (10), процесс заканчивается, и результатом будет то же входное слово. Если входное слово Vсодержит по крайней мере одно вхождение слова u, то на первом шаге будет применена формула (9), появится «решетка», после чего управление «перехватит» формула строки (7) (но если слово u однобуквенное, то может оказаться подходящей и формула (6)). Заметим, что в общем случае различные вхождения слова u могут пересекаться, и мы позволяем «решетки» аккуратно «перепрыгнуть» только через первую букву слова u. «Решетка» играет роль своего рода челнока и сканирует входное слово в поисках 2-го вхождения слова u. Если его нет, «решетка» добегает до конца входного слова и исчезает (формула (8)). Если же второе вхождение есть, то в определенный момент управление будет передано формуле (6), и второй челнок (двойная «решетка») отправляется искать третье вхождение слова u (строка (2)). Если оно не

обнаруживается, то, добежав до конца слова, двойная «решетка» превращается в «доллар», который стирает входное слово и исчезает сам (строки (4) и (5)). Условие выполнено, входное слово аннулировано. В противном случае челнок ## исчезает, и входное слово остается как было (формула (1)).

Студентам можно предложить в качестве несложного упражнения модифицировать схему так, чтобы НА аннулировал слова, в которых не менее трех вхождений заданного непустого слова.

2. Написать схему НА, распознающего (в том же смысле, как и в предыдущей задаче) такие слова в алфавите V, которые содержат хотя бы одно вхождение двух, отличных друг от друга(но возможно, что одно из них входит в другое) непустых слов.

В написанной ниже схеме параметр ξ пробегает алфавит V (т. е. $\xi \in V$), буквы $\alpha, \beta, \#, \nabla, @ \notin V$, $u, v \in V^+$ - слова, вхождения которых требуется найти.

$$\begin{cases} \nabla\xi \to \nabla \\ \xi\nabla \to \nabla \\ \nabla \to \cdot \\ \alpha v \to \nabla \\ \beta u \to \nabla \\ \alpha\xi \to \xi\alpha \\ \beta\xi \to \xi\beta \\ \alpha \to \cdot \\ \beta \to \cdot \\ \#u \to \alpha u \\ \#v \to \beta v \\ \#\xi \to \xi\# \\ \xi\# \to \cdot\xi \\ \# \to \cdot @ \\ \to \# \end{cases}$$

Некоторая тонкость состоит в том, что первый челнок (#) выставляется сразу перед всем входным словом. Это сделано для того, чтобы не упустить вложенного вхождения.

Примеры процессов работы:

$u = ba, v = aba$

1) $bbabbaba \mapsto \#bbabbaba \mapsto b\#babbaba \mapsto ba\alpha babbaba \mapsto^5$

$\mapsto^5 bbabba\alpha aba \mapsto bbabb\nabla \mapsto^5 \nabla \mapsto \cdot\lambda$

Заметим, что НА находит **первое** вхождение каждого слова.

2) $aba \mapsto \#aba \mapsto \beta aba \mapsto a\beta ba \mapsto a\nabla \mapsto \nabla \mapsto \cdot$

3) Работа с пустым словом: $\lambda \mapsto \# \mapsto \cdot @$

4) Работа с непустым словом, не содержащем какого-либо вхождения (в частности, ни одного):

$$ba \mapsto \#ba \mapsto \alpha ba \mapsto^2 ba\alpha \mapsto \cdot ba;$$
$$abb \mapsto \#abb \mapsto^3 abb\# \mapsto \cdot abb$$

Можно заметить, что расположение формул в 5-й и 6-й строках снизу друг относительно друга не существенно.

3. Написать схему НА, вычисляющего следующую функцию:

$$A_u(x) = \begin{cases} 1\#x, & \text{если непустое слово } u \text{ входит в } x \text{ ровно 2 раза} \\ 0\#x & \text{иначе} \end{cases},$$

где входное слово x есть слово в произвольно заданном алфавите V, а $\#, 0, 1 \notin V$.

$$\begin{cases} \xi\alpha \to \alpha\xi \;//\; \alpha \in \{0,1\}, \xi \in V \\ \alpha \to \cdot\alpha\# \\ \&u \to 0u \;//\; \& \notin V \\ \&\xi \to \xi\& \\ \& \to 1 \\ \$u \to u(1)\&u(2)...u(k) \;//\; k \geq 1, \$ \notin V \\ \$\xi \to \xi\$ \\ \$ \to 0 \\ u \to u(1)\$u(2)...u(k) \\ \to \cdot 0\# \end{cases}$$

Если во входном слове нет требуемых вхождений, то применяется самая нижняя формула, и перед словом возникает 0# (для пустого входного слова это будет результат). Иначе после первой буквы первого вхождения слова u выскакивает первый челнок ($), который запускается для поиска 2-го вхождения. Если его нет, «доллар» заменяется на 0, который по формулам самой верхней строки идет до упора влево и через решетку приставляется к входному слову слева (2-я строка сверху). При обнаружении 2-го вхождения «доллар» запускает аналогично второй челнок (амперсанд), который ищет 3-е вхождение (его можно заменить и двойным «долларом»). Если 3-е вхождение есть, амперсанд

превращается в 0 и точно так же, как и выше идет к началу входного слова и присоединяется к нему через «решетку» слева. При отсутствии 3-го вхождения амперсанд, добежав до конца входного слова, превращается в 1, которая так же, как и 0, бежит к левому краю и слева через решетку присоединяется к входному слову.

4. НА удвоения слова (лекционный пример).

Рассмотрим алгорифм *Double*, задаваемый схемой в алфавите $V_1 = V \cup \{\alpha, \beta\}$, причем $\alpha, \beta \notin V$:

$$\begin{cases} \alpha\xi \to \xi\beta\xi\alpha, \xi \in V \\ \beta\xi\eta \to \eta\beta\xi, \eta, \xi \in V \\ \beta \to \\ \alpha \to \cdot \\ \to \alpha \end{cases}$$

Можно показать, что $(\forall x \in V^*)(Double : x \models_{\cdot} xx)$, т.е. этот алгорифм удваивает любое входное слово в алфавите *V*.

Например, пусть *x* = *abca*. Тогда

Double: *abca* ⊢ α*abca* ⊢ *a*β*a*α*bca* ⊢ *a*β*ab*β*b*α*ca* ⊢ *a*β*ab*β*bc*β*c*α*a* ⊢ *a*β*ab*β*bc*β*ca*β*a*α ⊢ *ab*β*a*β*bc*β*ca*β*a*α ⊢ *ab*β*ac*β*b*β*ca*β*a*α ⊢ *ab*β*ac*β*ba*β*c*β*a*α ⊢ *abc*β*a*β*ba*β*c*β*a*α ⊢ *abc*β*aa*β*b*β*c*β*a*α ⊢ *abca*β*a*β*b*β*c*β*a*α ⊢^{4} *abcaabca*α ⊢$_{\cdot}$ *abcaabca* = *xx* .

Можно заметить следующее: ко входному слову применима только формула нижней строки рассматриваемой схемы. После ее применения посредством применения формул верхней строки каждая буква входного слова копируется и «копия» слева отмечается буквой β (буква α играет роль указателя и продолжает «бежать» вправо). После того, как указатель α пробежит все слово, начинают работать формулы второй строки, в результате применения которых каждая «копия» «добегает» до конца исходного слова (т.е. так, чтобы после нее не было «оригинальных» букв). Затем все буквы β и указатель α стираются.

Можно заметить также, что если вместо формулы $\beta \to$ поставить формулу $\beta\xi\alpha \to \xi\alpha$, то копия будет инвертирована, т.е. получится слово xx^R.

5. НА инвертирования входного слова

Зададим алгорифм Rv над алфавитом V такой схемой:

$$\begin{cases} \alpha\xi\eta \to \eta\alpha\xi, \xi,\eta \in V \\ \alpha\xi \to \beta\xi, \xi \in V \\ \beta\xi\beta \to \beta\xi \\ \alpha\beta \to \cdot \\ \alpha \to \cdot \\ \to \alpha \end{cases}$$

В этой схеме буквы α и β не принадлежат алфавиту *V*.

Пусть $x=x(1)x(2)\ldots x(m)$, $m > 0$, есть непустое слово в алфавите *V*.

В схеме алгорифма *Rv* к слову *x* применима только самая нижняя формула. После ее применения, при условии, что $m > 1$, будет применима одна из формул верхней строки. Эти формулы применяются до тех пор, пока слово αξ не окажется в конце слова:

$x(1)x(2)\ldots x(m) \vdash \alpha\, x(1)x(2)\ldots x(m) \vdash x(2)\alpha x(1)\ldots x(m) \vdash \ldots \vdash x(2)\ x(3)\ldots x(m)\alpha x(1)$.

Затем буква α «превратится» в букву β (вторая сверху строка схемы) и получится слово $x(2)\ x(3)\ldots x(m)\beta x(1)$. К нему опять-таки применима только самая нижняя формула; после ее применения будем иметь:

$x(2)\ x(3)\ldots x(m)\beta x(1) \vdash \alpha x(2)\ x(3)\ldots x(m)\beta x(1) \vdash$

$\vdash x(3)\ \alpha x(2)\ldots x(m)\beta x(1) \vdash \ldots \vdash x(3)\ldots x(m)\ \alpha x(2)\ \beta x(1) \vdash$

$\vdash x(3)\ldots x(m)\ \beta x(2)\ \beta x(1) \vdash x(3)\ldots x(m)\ \beta x(2)x(1).$

Потом буква $x(3)$ (если она есть) будет точно так же перенесена в конец слова и т. д. до тех пор, пока не получится слово $x(m)\ \beta\ x(m\text{-}1)\ldots x(2)x(1)$.

Далее:

$x(m)\ \beta\ x(m\text{-}1)\ldots x(2)x(1) \vdash \alpha x(m)\ \beta\ x(m\text{-}1)\ldots x(2)x(1) \vdash$

$\vdash \beta x(m)\ \beta\ x(m\text{-}1)\ldots x(2)x(1) \vdash \beta x(m)\ x(m\text{-}1)\ldots x(2)x(1)$

$\vdash \alpha\beta x(m)\ x(m\text{-}1)\ldots x(2)x(1) \vdash \cdot\ x(m)\ x(m\text{-}1)\ldots x(2)x(1).$

Итак, алгорифм *Rv* перерабатывает слово x, длина которого больше 1, в его инверсию x^R.

Для пустого и однобуквенного слова соответственно имеем:

$\lambda \vdash \alpha \vdash \cdot \lambda$

и

$a \vdash \alpha a \vdash \beta a \vdash \alpha\beta a \vdash \cdot a$.

Итак, мы можем утверждать, что $(\forall x \in V^*)(Rv(x) = x^R)$.

6. Удвоение через разделитель.

$$Double^{\$}: \begin{cases} \alpha\xi \to \xi\beta\xi\alpha \\ \beta\xi\eta \to \eta\beta\xi \\ \alpha \to \$ \\ \beta\xi\$ \to \$\xi \\ \$ \to \cdot\$ \\ \to \alpha \end{cases}$$

$(\xi, \eta \in V; \alpha, \beta, \$ \notin V)$

Можно показать, что для любого слова $x \in V^*$ $Double^{\$}(x) = x\$x$.

Арифметические алгорифмы

7. Сложение и умножение конструктивных натуральных чисел.

Схема сложения совсем простая:

$Add: \{\$0 \to \cdot$

Пара аргументов $011\ldots1\$011\ldots1$ перерабатывается в сумму $011\ldots1\underset{m}{}11\ldots1\underset{n}{}=011\ldots1$.

Схема умножения значительно сложнее.

$$Mult:\begin{cases} a1\to 1ba & (1)\\ a\to & (2)\\ b1\to 1b & (3)\\ 1\$0\to \$0a & (4)\\ 00\to 0 & (5)\\ \$0\to 0\$ & (6)\\ \$1\to \$ & (7)\\ \$b\to 1b & (8)\\ \$\to \cdot & (9) \end{cases}$$

Нужно показать, что $Mult(m\$n)=mn$.

Рассмотрим следующие случаи.

1) $m=0, n\geq 0$

$0\$01^n \mapsto_{(6)} 00\$1^n \mapsto_{(5)} 0\$1^n \mapsto_{(7)} \ldots_{n>0} \mapsto_{(7)} 0\$ \mapsto_{(9)} \cdot 0$.

2) $m>0, n=0$

$01^m\$0 \mapsto_{(4)} 01^{m-1}\$0a \mapsto_{(2)} 01^{m-1}\$0 \mapsto_{(4)} 01^{m-2}\$0a \mapsto_{(4),(2)} \ldots \mapsto 0\$0 \mapsto_{(6)} 00\$$
$\mapsto_{(5)} 0\$ \mapsto_{(9)} \cdot 0$

Итак, если хотя бы один аргумент равен нулю, то произведение равно нулю.

3) $m>0, n>0$

$01^m\$01^n \mapsto_{(4)} 01^{m-1}\$0a1^n \mapsto_{(1)} 01^{m-1}\$01ba1^{n-1} \mapsto_{(1)} \ldots \mapsto_{(1)} 01^{m-1}\$01\underset{n}{b\ldots1b}\, a \mapsto_{(2)} 01^{m-1}\$01\underset{n}{b\ldots1b}$
$\mapsto_{(3)} \ldots \mapsto_{(3)} 01^{m-1}\$01^n b^n \mapsto_{(4)} 01^{m-2}\$0a1^n b^n \mapsto_{(1)} 01^{m-2}\$01ba1^{n-1}b^n \mapsto_{(1)} \ldots \mapsto_{(1)} 01^{m-2}\$01\underset{n}{b\ldots1b}\, ab^n$
$\mapsto_{(2)} 01^{m-2}\$01\underset{n}{b\ldots1b}\, b^n \mapsto_{(3)} \ldots \mapsto_{(3)} 01^{m-2}\$01^n b^{2n} \mapsto \ldots \mapsto_{(4),(1),(2),(3)\ \text{еще}\ m-2\ \text{раз}} 0\$01^n b^{mn}$
$\mapsto_{(6)} 00\$1^n b^{mn} \mapsto_{(5)} 0\$1^n b^{mn} \mapsto_{(7)} \ldots \mapsto_{(7)} 0\$b^{mn} \mapsto_{(8)} \ldots \mapsto_{(8)} 01^{mn}\$ \mapsto_{(9)} \cdot 01^{mn} = mn$

Доказано. См. Б.А. Кушнер. Лекции по конструктивному математическому анализу. – М.: Наука, 1973. – С. 61-63.

8. Усеченное вычитание

$$\underset{\cdot}{-}\begin{cases}1\$01 \to \$0 & (1)\\ \$01 \to \$0 & (2)\\ \$0 \to \cdot & (3)\end{cases}$$

Легко показать, что $\underset{\cdot}{-}(m\$n) = \begin{cases} m-n, \text{ при } m > n \\ 0, \text{при } m \le n \end{cases}$

9. Распознавание нестрогого неравенства.

$$MoreEq: \begin{cases} 1\$01 \to \$0 \\ 0\$01 \to 0\$a \,/\, /a \notin \{0,1\} \\ 1\$0 \to \$0 \\ 0\$0 \to \cdot 1 \\ a1 \to a \\ \$a \to \cdot \end{cases}$$

Также нетрудно проверить, что $MoreEq(m\$n) = \begin{cases} 1, m \ge n \\ 0, m < n \end{cases}$.

9а. Более простое решение:

$$MoreEq: \begin{cases} 1\$01 \to \$0 \\ 11 \to 1 \\ 01\$0 \to \cdot 1 \\ 0\$0 \to \cdot 1 \\ 0\$01 \to \cdot 0 \end{cases}$$

Меняя 2-ю снизу формулу на $0\$0 \to \cdot 0$, получим НА распознающий строгое неравенство $m > n$.