

ЭФФЕКТИВНЫЕ АЛГОРИТМЫ ПОСТРОЕНИЯ ТЕРМОВ МИНИМАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ

Дадеркин Д.О.
Кафедра информатики

Поступила в редакцию 07.11.2016, после переработки 03.12.2016.

Рассматривается задача построения термов минимальной вычислительной сложности программой, содержащей только присваивания и конечное число переменных, работающей на свободном однопорожденном группоиде с сигнатурой, состоящей из символа двуместной операции и одного порождающего элемента. Рассматриваются такие термы, что листья соответствующих им полных бинарных деревьев помечены попарно различными подтермами. Приводится эффективный алгоритм такого построения и доказывается, что для любого натурального m можно построить терм высоты h в $h - m$ переменных.

Ключевые слова: бинарное дерево, вычислительная сложность, однопорожденный группоид, переменная, присваивание, терм, эффективный алгоритм.

Вестник ТвГУ. Серия: Прикладная математика. 2016. № 4. С. 21–33.

1. Введение

Тюрын в [1] приводит утверждение, дающее оценку вычислительной сложности $cc(t)$ терма t , зависящую от высоты h этого терма: $cc(t) \geq h + 1$. Кфури в [2], опираясь на [1], получил интересные результаты, касающиеся проблем сложности. М.А. Тайцлин и И.Х. Мусикаев в [3] привели пример, опровергающий по крайней мере на единицу утверждение из [1], построив в двух переменных терм t высоты 2, доказав тем самым, что для этого терма $cc(t) = h$, и показав, что оценка Тюрына верна лишь при определенных ограничениях, накладываемых на терм. М.А. Тайцлин задал вопрос: действительно ли оценка вычислительной сложности терма в формулировке Тюрына отличается от истинной не только на 1, как в построенном в [3] примере? Отвечая на этот вопрос, докажем, что для любого натурального m с использованием $h - m$ переменных можно построить удовлетворяющий формулировке Тюрына терм $t(m)$ высоты h , то есть $cc(t) = h - m$, и приведем эффективный алгоритм такого построения.

Пусть \mathbb{X} – конечное множество переменных, a – единственная (с точностью до имени) константа. Терм t – либо константа a , либо переменная $x \in \mathbb{X}$, либо выражение вида $(t_1 \cdot t_2)$, где t_1 и t_2 – термы. Термы образуют множество термов T . Как обычно, термы сравниваются как слова.

Пусть B – множество всех конечных бинарных деревьев. Введем на множестве B бинарную операцию $\circ : B^2 \rightarrow B$ сцепления деревьев: скажем, что если $b_1 \in B$ и $b_2 \in B$, то $b = b_1 \circ b_2 \in B$, b_1 – левое поддереву дерева b , b_2 – правое.

Интерпретируем в B константу a как дерево, состоящее только из одной вершины, помеченной a , операцию умножения \cdot термов – как операцию \circ сцепления деревьев, терм $t = (t_1 \cdot t_2)$ – как дерево $b = b_1 \circ b_2$, помеченное t , при этом терму t_1 соответствует дерево b_1 , помеченное t_1 , терму t_2 – дерево b_2 , помеченное t_2 . Такая интерпретация помечает вершины соответствующими значениями и превращает B в алгебру $\mathbf{B} = \langle B, \circ, a \rangle$.

Далее, говоря о высоте терма, пути в терме, подтерме будем понимать соответствующие термины в помеченном в соответствии с интерпретацией дереве.

2. Базовые понятия и обозначения

Пусть \mathbb{N} – множество натуральных чисел, \mathbb{X} – конечное множество переменных. Выражения вида $x \leftarrow a$, $x \leftarrow y \cdot z$, $x \leftarrow y$ назовем присваиваниями. Здесь $x, y, z \in \mathbb{X}$, a – константа. Присваивание является программой. Если S_1 и S_2 – программы, то $S_1; S_2$ – программа.

Обычным образом вводится состояние как частичное отображение множества переменных \mathbb{X} в множество термов T . Значение переменной $x \in \mathbb{X}$ на состоянии σ есть σx .

Если S есть $x \leftarrow a$ и σx – некоторое состояние перед выполнением S , то $S(\sigma)(x)$ есть a . Если S есть $x \leftarrow y \cdot z$ и $\sigma y = t_1$, $\sigma z = t_2$ перед выполнением S , то $S(\sigma)(x)$ есть $(t_1 \cdot t_2)$. Если S есть $x \leftarrow y$ и $\sigma y = t_1$ перед выполнением S , то $S(\sigma)(x)$ есть t_1 . При этом программные переменные, отличные от x , своих значений не меняют.

Если S есть $S_1; S_2$, то $S(\sigma)$ есть $S_2(S_1(\sigma))$.

Программа S вычисляет терм $t \in T$ в переменной $x \in \mathbb{X}$, если $S(\sigma)(x) = t$.

Если терм $t \in T$ вычисляется программой S , в которой используется не более n переменных, то скажем, что вычислительная сложность этого терма не превосходит n . *Вычислительной сложностью $cc(t)$ терма t называется наименьшее из таких n .*

Пусть \bar{S}_i – такая программа, что $\bar{S}_i(\sigma)(v_i) = t_i$, $cc(t_i) = i$, причем, для $1 \leq k \leq i$ первым присваиванием в \bar{S}_k является $v_1 \leftarrow a$, если $k = i$, и $v_1 \leftarrow v_{k+1} \cdot v_{k+1}$, если $1 \leq k < i$. Определим программу S_i : для $i > 1$ $S_i = \{\bar{S}_i; \bar{S}_{i-1}; \dots, \bar{S}_1\}$, для $i = 1$ $S_1 = v_1 \leftarrow a$.

Пример 1. Для $i = 3$ $S_3 = \{\bar{S}_3; \bar{S}_2; \bar{S}_1\}$, и в \bar{S}_3 есть присваивание $v_1 \leftarrow a$, в \bar{S}_2 есть присваивание $v_1 \leftarrow v_3 \cdot v_3$, в \bar{S}_1 есть присваивание $v_1 \leftarrow v_2 \cdot v_2$.

Таким образом, программа S_i вычисляет в переменных v_i, \dots, v_1 термы t_i, \dots, t_1 высот $h(t_i), \dots, h(t_1)$ вычислительной сложности $cc(t_j) \leq i$, $1 \leq j \leq i - 1$, соответственно, и $\bar{S}_i(\sigma)(v_i) = t_i$, $cc(t_i) = i$.

Пусть корень дерева b помечен в соответствии с введенной интерпретацией термом t , и $t \neq a$. Тогда ясно, что все остальные помеченные вершины этого дерева есть подтермы терма t .

Далее для краткости вместо «корень дерева b помечен в соответствии с введенной интерпретацией термом t » будем говорить «терм t ».

Определение 1. Скажем, что t' – подтерм терма t , если $x, y \in \mathbb{X}$, $S(\sigma)(x) = t$ и $S(\sigma)(y) = t'$.

Пример 2. Пусть программа $S : x \leftarrow a; y \leftarrow x \cdot x; x \leftarrow x \cdot y$, тогда $S(\sigma)(x)$ есть терм $t = (a \cdot (a \cdot a))$, $S(\sigma)(y)$ есть подтерм t' терма t , и $t' = (a \cdot a)$.

Определение 2. Назовем i -ым слоем $i(t)$ терма $t \in T$ высоты n совокупность подтермов терма t , определяемых всеми исходящими от корня терма t путями длины $n - i$.

Пример 3. Для терма $t = ((a \cdot a) \cdot a)$ высоты $n = 2$ слой $0(t) = \{a, a\}$, $1(t) = \{(a \cdot a), a\}$, $2(t) = \{((a \cdot a) \cdot a)\}$.

Определение 3. Назовем правильным слой $i(t)$ терма t высоты n , если этот слой есть $n(t)$, либо все входящие в него подтермы попарно различны.

Пример 4. В терме t из примера 2.3 слой $1(t)$ – правильный, $0(t)$ – нет.

Определение 4. Слои $i(t)$ и $j(t)$ попарно независимы, если $i(t)$ и $j(t)$ – правильные, а подтермы, входящие в $i(t)$ и в $j(t)$, попарно различны.

Слой $i(t)$ независим, если $i = 0$ и $0(t)$ – правильный слой, либо $i > 0$ и для любого $j < i$ слои $i(t)$ и $j(t)$ попарно независимы.

Ясно, что если слой $i(t)$ независим, то в нижележащих слоях нет термов, входящих в i -тый слой.

Пример 5. Пусть $X2$ означает $(a \cdot a)$, $X3 - (a \cdot X2)$, $X4 - (X3 \cdot X3)$, $X5 - (X4 \cdot X4)$, $X6 - (X4 \cdot X5)$, $X7 - (X3 \cdot X6)$. Тогда в терме, помеченном $X7$ в этих обозначениях, высота $h(X7) = 6$, слои $4(X7) = \{a, X2, X4, X5\}$, $5(X7) = \{X3, X6\}$, и слои $4(X7)$ и $5(X7)$ попарно независимы: в $4(X7)$ все термы различны, в $5(X7)$ все термы различны, в $4(X7)$ нет подтермов $X3$ и $X6$, а в $5(X7)$ нет подтермов $a, X2, X4, X5$.

Определение 5. Пусть t – терм высоты n . Надтермом $\overline{t_k}$ высоты $k \leq n$ терма t называется такой терм, что $0(\overline{t_k}) = (n - k)(t)$.

Следующий пример показывает, что у такого надтерма $\overline{t_k}$ метки всех листьев совпадают с соответствующими подтермами слоя $(n-k)$ терма t .

Пример 6. Пусть $X7$ – терм из примера 2.5. Для $k = 0$ рассмотрим $\overline{X7_0}$. По определению 5, $0(\overline{X7_0}) = (6 - 0)(X7) = 6(X7)$. Таким образом, надтерм $\overline{X7_0}$ – это терм $X7$, которому соответствует дерево высоты 0 с листьями, помеченными подтермами из слоя $6(X7)$, т.е. корень дерева, помеченный $X7$.

Для $k = 1$ рассмотрим $\overline{X7_1}$. По определению 5, $0(\overline{X7_1}) = (6 - 1)(X7) = 5(X7)$. Таким образом, надтерм $\overline{X7_1}$ – это терм $X7$, которому соответствует дерево высоты 1 с корнем, помеченным $X7$, и сыновьями $X7$ являются $X3$ и $X6$, т.е. подтермы из слоя $5(X7)$ терма $X7$, причем, $X3$ и $X6$ – листья для $\overline{X7_1}$.

Для $k = 2$ рассмотрим $\overline{X7_2}$. По определению 5, $0(\overline{X7_2}) = (6 - 2)(X7) = 4(X7)$. Таким образом, надтерм $\overline{X7_2}$ – это терм $X7$, которому соответствует дерево высоты 2 с корнем, помеченным $X7$, сыновьями которого являются вершины, помеченные $X3$ и $X6$, сыновья $X3$ есть a и $X2$, сыновья $X6$ – $X4$ и $X5$, при этом $a, X2, X4$ и $X5$ являются подтермами из слоя $4(X7)$ терма $X7$, т.е. листьями $\overline{X7_2}$.

Для $k = 3$ рассмотрим $\overline{X7_3}$. По определению 5, $0(\overline{X7_3}) = (6 - 3)(X7) = 3(X7)$. Таким образом, надтерм $\overline{X7_3}$ – это терм $X7$, которому соответствует дерево высоты $X3$ с корнем, помеченным $X7$, сыновьями которого являются вершины, помеченные $X3$ и $X6$, сыновья $X3$ есть a и $X2$, при этом a является листом по построению дерева $X7$, сыновья $X6$ – $X4$ и $X5$, а $X2$, $X4$ и $X5$ имеют сыновей $\{a, a\}$, $\{X3, X3\}$, $\{X4, X4\}$, которые являются подтермами из слоя $3(X7)$ термина $X7$, т.е. листьями $\overline{X7_3}$.

Ясно, что надтермам $\overline{X7_0}$, $\overline{X7_1}$, $\overline{X7_2}$ соответствуют полные бинарные деревья высот $0, 1$ и 2 , соответственно, а $\overline{X7_3}$ не является полным бинарным деревом: пути длины 3 приводят только к 6 листьям, помеченным $a, a, X3, X3, X4, X4$, а путь до самого левого листа a имеет длину 2 .

Пусть программа S_k вычисляет терм t_k в переменной v_k : $S(\sigma)(v_k) = t_k$, и $\sigma(v_1) = b$ для некоторого $b \in T$. Пусть m – длина минимального пути из корня t_k до b , тогда рассмотрим такой надтерм $\overline{t_k}$, что $h(\overline{t_k}) = m$. Далее под термом t будем понимать $\overline{t_k}$, и $h(t) = h(\overline{t_k})$. Ясно, что таким образом интерпретируемый терм t есть полное бинарное дерево высоты $h(t)$ с помеченными в соответствии с введенной интерпретацией вершинами.

Определение 6. Пусть t – терм высоты n . Терм t – l -правильный, если $l(t)$ – независимый слой. Терм t – правильный, если для любого l : $0 \leq l \leq n + 1$ терм t – l -правильный.

Определение 7. Пусть T – лес из k деревьев r_j : $T = \bigcup_{j=1}^k r_j$. Терм t независим в лесу T из k деревьев, если t – 0 -правильный терм и слой $0(t)$ попарно независим со всеми слоями $0(r_j)$, $1 \leq j \leq k$. Если ясно, о каком лесе идет речь, то терм t назовем независимым, если он независим в данном лесе.

Определение 8. Лес T – правильный, если все деревья в нем независимы.

Определение 9. Пусть t_i, t_j – термы высот $h(t_i)$ и $h(t_j)$, соответственно. Скажем, что между термами t_i и t_j существует разрыв, если $h(t_i) > h(t_j) + 1$. Глубина разрыва $n = h(t_i) - h(t_j) - 1$.

Определение 10. Пусть совокупность термов $\langle t_j \rangle = \{t_j, t_{j-1}, \dots, t_1\}$. Для $t_i \in \langle t_j \rangle$, $1 \leq i \leq j$, определим вычислительную сложность совокупности термов $\langle t_j \rangle$:

$$cc(\langle t_j \rangle) = \max_{1 \leq i \leq j} cc(t_i).$$

3. Увеличение высоты независимого надтерма

Лемма 1. Пусть $i, l \in \mathbb{N}$, $i > 0$, программа S_i вычисляет термы r_i, \dots, r_1 : $S_i(\sigma)(v_j) = r_j$, $1 \leq j \leq i$, слои $0(r_j)$ – правильные, $cc(r_j) \leq i$, $1 \leq j \leq i-1$, $cc(r_i) = i$, в r_i существует независимый в лесу $\{r_i, \dots, r_1\}$ надтерм $\overline{r_i}$ высоты $h(\overline{r_i})$.

Тогда существует такая программа S_{i+l} , вычисляющая термы $r'_{i+l}, \dots, r'_i, \dots, r'_1$: $S_{i+l}(\sigma)(v_j) = r'_j$, $1 \leq j \leq i+l$ высот $h(r'_{i+l}), \dots, h(r'_i), \dots, h(r'_1)$, что

$$(1.1) \quad h(r'_{i+l}) = h(r_i) + l,$$

$$(1.2) \quad cc(r'_{i+l}) = cc(r_i) + l,$$

$$(1.3) \quad h(r'_j) = h(r_j), j = 1, \dots, i,$$

$$(1.4) \quad cc(r'_j) = cc(r_j), j = 1, \dots, i,$$

$$(1.5) \quad \text{слой } 0(r'_j) \text{ - правильный, } j = 1, \dots, i + l,$$

$$(1.6) \quad \text{в } r'_{i+l} \text{ существует независимый в лесу } \{r'_{i+l}, \dots, r'_1\} \text{ надтерм } \overline{r'_{i+l}} \text{ высоты } h(\overline{r'_{i+l}}) = h(\overline{r_i}) + l.$$

Доказательство. Индукция по количеству l дополнительных программных переменных. Пусть выполнена программа S_i .

Базис индукции. $l = 1$. Рассмотрим программу

$$S_{i+1} = \{v_{i+1} \leftarrow v_i; v_1 \leftarrow v_i \cdot v_i; S_i; v_{i+1} \leftarrow v_i \cdot v_{i+1};\}.$$

Пусть r_i - результат работы программы S_i , полученный в переменной $v_i : S_i(\sigma)(v_i) = r_i, r'_{i+1}$ - результат работы программы S_{i+1} , полученный в переменной $v_{i+1} : S_{i+1}(\sigma)(v_{i+1}) = r'_{i+1}$.

Ясно, что $cc(r'_{i+1}) = cc(r_i) + 1, h(r'_{i+1}) = h(r_i) + 1$, и тогда во множестве термов $\langle r'_{i+1} \rangle$, полученных программой S_{i+1} , существует надтерм $\overline{r'_{i+1}}$ высоты $h(\overline{r'_{i+1}}) = h(\overline{r_i}) + 1$. Независимость $\overline{r'_{i+1}}$ следует из построения S_{i+1} , пункт (1.6) доказан. (1.1) - (1.5) очевидны. Базис обоснован.

Индуктивный переход. Пусть при $l = k$ существует такая программа S_{i+k} , использующая переменные $v_{i+k}, \dots, v_1 : S_{i+k}(\sigma)(v_j) = r'_j, j = 1, \dots, i + k$, что (1.1) - (1.6) верно. Докажем для $l = k + 1$. Построим программу S_{i+k+1} .

Обозначим S''_{i+k} программу, отличающуюся от существующей по индукционному предположению программы S_{i+k} тем, что в ней во всех присваиваниях переменная v_{i+k} заменена на переменную v_{i+k+1} , не встречавшуюся ранее в S_{i+k} . Тогда программа S''_{i+k} , такая, что $S''_{i+k}(\sigma)(v_j) = r''_j, j = 1, \dots, i + k - 1$, вычисляя в v_{i+k+1} терм $r''_{i+k} : S''_{i+k}(\sigma)(v_{i+k}) = r''_{i+k}$, не изменяет высот и сложности термов $r''_{i+k-1}, \dots, r''_1$, вычисляемых в оставшихся переменных v_{i+k-1}, \dots, v_1 , то есть

$$(1) \quad h(r''_j) = h(r_j), cc(r''_j) = cc(r_j), j = 1, \dots, i + k - 1,$$

$$h(r''_{i+k}) = h(r_{i+k}), cc(r''_{i+k}) = cc(r_{i+k}).$$

Построим программу

$$S'_{i+k+1} = \{S_{i+k}; v_1 \leftarrow v_{i+k} \cdot v_{i+k}; S''_{i+k}; v_{i+k+1} \leftarrow v_{i+k} \cdot v_{i+k+1}\}.$$

Вычисляемые этой программой термы обладают следующими свойствами:

(2) Надтерм $\overline{r''_{i+k}}$ высоты $h(\overline{r''_{i+k}})$ независим в лесу $\{r''_{i+k}, \dots, r''_1\}$ по индукционному предположению, независим в терме r_{i+k} по построению S'_{i+k+1} .

(3) Надтерм $\overline{r_{i+k}}$ высоты $h(r_{i+k})$ независим в терме r_{i+k} по индукционному предположению, независим в лесу $\{r''_{i+k}, \dots, r''_1\}$ по построению S'_{i+k+1} .

(4) Слои $0(\overline{r''_{i+k}})$ и $0(\overline{r_{i+k}})$ попарно независимы по построению S'_{i+k+1} .

$$(5) \quad S_{i+k+1}(\sigma)(v_j) = r_j, j = 1, \dots, i + k - 1.$$

$$S_{i+k+1}(\sigma)(v_{i+k}) = r_{i+k}.$$

$$S'_{i+k+1}(\sigma)(v_j) = r''_j, j = 1, \dots, i + k - 1,$$

то есть (1) справедливо для S'_{i+k+1} .

$$(6) \quad S'_{i+k+1}(\sigma)(v_{i+k+1}) = r''_{i+k+1}, \text{ и из (2)-(4) следует, что надтерм}$$

$$\overline{r''_{i+k+1}} = (\overline{r_{i+k}} \cdot \overline{r''_{i+k}}) \text{ независим в лесу } \{r_{i+k}, r''_{i+k+1}, r''_{i+k}, \dots, r''_1\}, \text{ и}$$

(7) в терме r''_{i+k+1} слой $0(r''_{i+k+1})$ - правильный. Очевидно, что

$$(8) \quad h(r''_{i+k+1}) = (h(r_{i+k}) + 1,$$

$$(9) \quad cc(r''_{i+k+1}) = (cc(r_{i+k}) + 1,$$

$$(10) \quad h(\overline{r''_{i+k+1}}) = h(\overline{r_{i+k}} + 1).$$

Итак, программа $S_{i+k+1} = \{S'_{i+k+1}; S_{i+k}\}$ вычисляет терм r''_{i+k+1} с надтермом $\overline{r''_{i+k+1}}$: $S_{i+k+1}(\sigma)(v_{i+k+1}) = r''_{i+k+1}$, и термы $S_{i+k+1}(\sigma)(v_j) = r'''_j, j = i+k, \dots, 1$, удовлетворяющие (1.1) - (1.6):

(1.1) и (1.2) следуют из (8) и (9), (1.3) и (1.4) – из индукционного предположения для S_{i+k} , (1.5) – из индукционного предположения для $r'''_{i+k}, \dots, r'''_1$ и (7), (1.6) – из (6) и построения S_{i+k+1} . Лемма доказана. \square

Следствие 1. Пусть термы r'_{i+l}, \dots, r'_i вычислены программой S_{i+l} , тогда

$$(1.1.1) \quad h(r'_{i+j}) = h(r'_{i+j-1}) + 1, j = 1, \dots, l,$$

$$(1.1.2) \quad cc(r'_{i+j}) = cc(r'_{i+j-1}) + 1, j = 1, \dots, l.$$

Доказательство очевидно следует из (1.1) - (1.6) Леммы 1 и построения S_{i+l} .

Из (1.5) получаем

Следствие 2. Пусть $S_{i+l}(\sigma)(v_j) = r'_j, j = 1, \dots, l$. Тогда каждый из термов $r'_j, j = 1, \dots, i+l$, вычисленных программой S_{i+l} , независим в лесу

$$\left\{ \bigcup_{k=1}^{i+l} r'_k \right\} \setminus r'_j.$$

4. Ликвидация разрыва конечной глубины

Обозначим $[v_{j+1-i} \leftarrow v_{j-i}]_{i=0}^l$ последовательность присваиваний $v_{j+1} \leftarrow v_j; v_j \leftarrow v_{j-1}; \dots; v_{j+1-l} \leftarrow v_{j-l}$.

Лемма 2. Пусть $i, j, k, p \in \mathbb{N}, p > 0$, программа R_j вычисляет совокупность термов $\langle t_j \rangle: R_j(\sigma)(v_i) = t_i, 1 \leq i \leq j$ таких, что

$$(2.1) \quad h(t_{j-k}) = h(t_{j-k-1}) + 1 + p,$$

$$(2.2) \quad h(t_{j-i}) = h(t_{j-i-1}) + 1, i \neq k, 0 \leq i \leq j-2,$$

$$(2.2.1) \quad \text{все термы } t_i \text{ независимы, } 1 \leq i \leq j.$$

Тогда существует такая программа D_{j+p} , вычисляющая совокупность термов $\langle t'_{j+p} \rangle: D_{j+p}(\sigma)(v_i) = t'_i, 1 \leq i \leq j+p$ таких, что

$$(2.3) \quad h(t'_{j+p-i}) = h(t'_{j+p-i-1}) + 1, 0 \leq i \leq p+j-2,$$

$$(2.4) \quad cc(\langle t'_{j+p} \rangle) = cc(t_j) + p,$$

$$(2.5) \quad h(t'_i) = h(t_i), 1 \leq i \leq j-k-1,$$

$$h(t'_{i+p}) = h(t_i), j-k \leq i \leq j,$$

$$(2.6) \quad h(t'_{j-k+p-1}) = h(t_{j-k-1}) + p,$$

$$(2.7) \quad \text{все термы } t'_i \text{ независимы, } 1 \leq i \leq j+p.$$

Доказательство. Пусть программа R_j выполнена. Построим программу D_{j+p} , вычисляющую совокупность термов $\langle t'_{j+p} \rangle$, удовлетворяющих (2.3) - (2.7).

Из (2.1) следует, что k определяет в последовательности термов, вычисленных программой R_j , место разрыва глубины p .

Программой $DG_p = \{v_1 \leftarrow v_j \cdot v_j; [S_{(j-k-1)+l}; v_1 \leftarrow v_{j-k-1+l} \cdot v_{j-k-1+l}]_{l=1}^p\}$ заполним разрыв, вычислив p необходимых термов. Здесь $S_{(j-k-1)+l}$ – программа, существующая по лемме 1.

Программой $DR_{j-k-1} = R_j^p$ вновь построим термы до и после точки разрыва. Здесь R_j^p – программа, отличная от R_j только тем, что в ней все переменные v_i

заменены на v_{i+p} , $j - k \leq i \leq j$. Покажем, что $D_{j+p} = \{DG_p; DR_{j-k-1}\}$ является искомой программой.

Индукция по глубине разрыва p .

Пусть лес $\langle t_j \rangle$, вычисленный R_j , удовлетворяет (2.1) и (2.2), α – начальное состояние для R_j , $R_j(\alpha)(v_j) = t_j$, $\sigma = R_j(\alpha)$.

Базис индукции. $p = 1$. Между термами t_{j-k} и t_{j-k-1} , находящимися в переменных v_{j-k} и v_{j-k-1} , существует разрыв глубины 1.

Выполним $D_{j+1} = \{DG_1; DR_{j-k-1}\}$. Здесь $\sigma(v_j) = R_j(\alpha)(v_j)$,

$DG_1 = \{v_1 \leftarrow v_j \cdot v_j; S_{j-k}; v_1 \leftarrow v_{j-k} \cdot v_{j-k}\}$.

Пусть $\tau = DG_1(\sigma)$, обозначим $t_i^1 = \tau(v_i)$, $1 \leq i \leq j - k$, и после выполнения DG_1 получим $(\tau)(v_{j-k}) = t_{j-k}^1$, $(\tau)(v_1) = t_1^1$. В DG_1 использована S_{j-k} , существующая по лемме 1. Тогда из (1.1) получаем:

(1) $h(t_{j-k}^1) = h(t_{j-k-1}^1) + 1$, из (2.2) и (1.3) $h(t_i^1) = h(t_i)$, $1 \leq i \leq j - k - 1$, отсюда и из (1):

(2) $h(t_{j-k}^1) = h(t_{j-k-1}) + 1$.

Заметим, что $DG_1(\sigma)(v_1) = t_1^1$.

Выполнив теперь DR_{j-k-1} , получим $t_i^2 = DR_{j-k-1}(\tau)(v_i)$, $1 \leq i \leq j+1$, $i \neq j-k$.

По построению DR_{j-k-1}

(3) $h(t_i^2) = h(t_i)$, $1 \leq i \leq j - k - 1$, $h(t_{i+1}^2) = h(t_i)$, $j - k \leq i \leq j$.

Окончательно, $t_i' = D_{j+1}(\sigma)(v_i)$, $1 \leq i \leq j + 1$,

$t_{j-k}' = t_{j-k}^1$,

$t_i' = t_i^2$, $1 \leq i \leq j + 1$, $i \neq j - k$. Из (2) следует, что $h(t_{j-k}') = h(t_{j-k-1}) + 1$, следовательно, (2.6) выполнено.

(2.3) и (2.4) очевидны, (2.5) следует из (3).

Из (2.2.1) и построения DG_1 следует, что термы t_i^1 независимы, $1 \leq i \leq j - k$. Из построения DR_{j-k-1} следует, что термы t_i^2 независимы, $1 \leq i \leq j + 1$, $i \neq j - k$. Из построения D_{j+1} следует, что все термы t_i' , $1 \leq i \leq j + 1$ независимы, следовательно, (2.7) верно. Базис обоснован.

Индуктивный переход. Пусть совокупность термов $\langle t_j \rangle$, вычисленная R_j , удовлетворяет (2.1) и (2.2), и между термами t_{j-k} и t_{j-k-1} , находящимися в переменных v_{j-k} и v_{j-k-1} , существует разрыв глубины p , программа $D_{j+p} = \{DG_p; DR_{j-k-1}\}$ ликвидирует этот разрыв, вычисляя совокупность термов $\langle t_{j+p}' \rangle$, удовлетворяющих (2.3) - (2.7).

Итак, пусть α – начальное состояние для R_j , $R_j(\alpha)(v_j) = t_j$, $\sigma = R_j(\alpha)$, а лес $\langle t_j \rangle$, вычисленный R_j , таков, что между термами t_{j-k} и t_{j-k-1} , находящимися в переменных v_{j-k} и v_{j-k-1} , существует разрыв глубины $p + 1$.

Построим программу $D_{j+p+1} = \{DG_{p+1}; DR_{j-k}\}$, вычисляющую совокупность термов $\langle t_{j+p+1}' \rangle$, удовлетворяющих (2.3) - (2.7) при разрыве глубины $p + 1$.

Положим $DG_{p+1} = \{v_1 \leftarrow v_j \cdot v_j; [S_{(j-k-1)+l}; v_1 \leftarrow v_{j-k-1+l} \cdot v_{j-k-1+l}]_{l=1}^{p+1}\} = \{v_1 \leftarrow v_j \cdot v_j; [S_{(j-k-1)+l}; v_1 \leftarrow v_{j-k-1+l} \cdot v_{j-k-1+l}]_{l=1}^p; [S_{(j-k-1)+p+1}; v_1 \leftarrow v_{j-k-1+p+1} \cdot v_{j-k-1+p+1}]\} = \{DG_p; S_{j-k+p}; v_1 \leftarrow v_{j-k+p} \cdot v_{j-k+p}\}$.

Программа DG_p , существующая по индукционному предположению, ликвидирует разрыв глубины p , программа S_{j-k+p} , существующая по Лемме1, построит в v_{j-k+p} независимый терм высоты на 1 больше, чем терм максимальной высоты, построенный DG_p , ликвидируя тем самым разрыв глубины $p + 1$.

Итак, положим $\tau_1 = DG_p(\sigma)$, обозначим $t_i^1 = \tau_1(v_i)$, $1 \leq i \leq j - k + p - 1$, и $t_{j-k+p-1}^1 = \tau_1(v_{j-k+p-1})$, $t_1^1 = \tau_1(v_1)$.

По индукционному предположению, для термов $\langle t_i^1 \rangle, 1 \leq i \leq j - k + p - 1$, выполнены (2.3) - (2.7).

Пусть $\tau_2 = \{S_{j-k+p}; v_1 \leftarrow v_{j-k+p} \cdot v_{j-k+p}\}(\tau_1)$, тогда $\tau_2(v_{j-k+p}) = t_{j-k+p}^2$, и легко видеть, что $\tau_2(v_{j-k+p-1}), \dots, \tau_2(v_1)$ таковы, что

$$(4) \ h(t_{j-k+p-1}^2) = h(t_{j-k+p-1}^1), \dots, h(t_1^2) = h(t_1^1),$$

то есть, для них выполнены (2.3) - (2.7). По (1.1.1) Следствия 1.1

$$h(t_{j-k+p}^2) = h(t_{j-k+p-1}^2) + 1, \text{ по (1.1.2) Следствия 1.1}$$

$cc(t_{j-k+p}^2) = cc(t_{j-k+p-1}^2) + 1$. Из (4) и индукционного предположения следует

$$(5) \ h(t_{j-k}^2) = h(t_{j-k-1}) + 1,$$

$$h(t_{j-k+p}^2) = h(t_{j-k+p-1}) + 1. \text{ Следовательно,}$$

$$(6) \ \langle t_{j-k+p}^2 \rangle \text{ удовлетворяют (2.3) - (2.7).}$$

Программа DR_{j-k} из DG_{p+1} отличается от DR_{j-k-1} из DG_p только номерами в именах переменных: $DR_{j-k} = R_j^{p+1}$, в R_j^{p+1} все переменные v_i заменены на $v_{i+p+1}, j - k \leq i \leq j$. Тогда после выполнения DR_{j-k} получим $\tau_3 = DR_{j-k}(\tau_2)$.

Обозначим $t_i^3 = DR_{j-k}(\tau_3)(v_i), \{1 \leq i \leq j - k - 1\} \cup \{j - k + p + 1 \leq i \leq j + p + 1\}$.

Из построения DR_{j-k} и R_j^{p+1} следует, что

$$(7) \ h(t_i^3) = h(t_i), cc(t_i^3) = cc(t_i), 1 \leq i \leq j - k - 1,$$

$$h(t_{i+p+1}^3) = h(t_i), cc(t_i^3) = cc(t_{i+p+1}), j - k \leq i \leq j.$$

Из построения D_{j+p+1} следует, что

$$D_{j+p+1}(\sigma)(v_i) = DR_{j-k}(DG_{p+1})(\sigma)(v_i), 1 \leq i \leq j + p + 1.$$

Обозначим $t'_i = D_{j+p+1}(\sigma)(v_i), 1 \leq i \leq j + p + 1$, тогда

$$t'_i = t_i^3, \{1 \leq i \leq j - k - 1\} \cup \{j - k + p + 1 \leq i \leq j + p + 1\},$$

$$t'_i = t_i^2, j - k - 1 < i \leq j - k + p.$$

Тогда для $\langle t'_{j+p+1} \rangle$ получаем:

$$(8) \ h(t'_i) = h(t_i^3) = h(t_{i-1}^3) + 1 = h(t'_{i-1}) + 1, j - k + p + 1 \leq i \leq j + p + 1$$

$$(9) \ h(t'_i) = h(t_i^2) = h(t_{i-1}^2) + 1 = h(t'_{i-1}) + 1, j - k + p \leq i \leq j - k$$

$$(10) \ h(t'_i) = h(t_i^3) = h(t_{i-1}^3) + 1 = h(t'_{i-1}) + 1, 1 \leq i \leq j - k - 1.$$

Из (8) - (10) следует истинность (2.3), (2.4) и (2.5) очевидны. Легко заметить,

что

$$h(t'_{j-k+p}) = h(t_{j-k+p}^2) = h(t_{j-k}^2) + p = h(t'_{j-k}) + p,$$

$$h(t'_{j-k}) = h(t_{j-k}^2) = h(t_{j-k-1}^2) + 1 = h(t'_{j-k-1}) + 1, \text{ тогда}$$

$$h(t'_{j-k+p}) = h(t'_{j-k-1}) + p + 1, \text{ но } h(t'_{j-k+p}) = h(t'_{j-k+p-1}) + 1, \text{ и, учитывая (7),}$$

получаем, что (2.6) верно.

(2.7) следует из индукционного предположения и построения D_{j+p+1} . Лемма доказана. \square

5. Ликвидация конечного числа разрывов

Лемма 3. Пусть $i, i_1, j, l, K_l, P_l \in \mathbb{N}, P_l > 0, 1 \leq l \leq i_1$, программа RN_j вычисляет термы $t_j, \dots, t_1 : RN_j(\sigma)(v_i) = t_i, 1 \leq i \leq j$ такие, что в совокупности термов $\langle t_j \rangle = \{t_j, \dots, t_{j-K_{i_1}}, t_{j-K_{i_1}-1}, \dots, t_{j-K_1}, t_{j-K_1-1}, \dots, t_1\}$ имеется i_1 разрыв K_l глубины $P_l > 0, 1 \leq l \leq i_1$.

Обозначим

$$m_u = \sum_{l=1}^u (P_l), m_0 = 0.$$

Вычисляемые программой RN_j термы $\langle t_j \rangle$ обладают следующими свойствами:

$$(3.1) \quad h(t_{j-K_l}) = h(t_{j-K_{l-1}}) + 1 + P_l, 1 \leq l \leq i1,$$

$$(3.2) \quad h(t_{j-i}) = h(t_{j-i-1}) + 1, i \neq K_l, 0 \leq i \leq j-2, 1 \leq l \leq i1,$$

$$(3.2.1) \quad \text{все термы } t_i \text{ независимы, } 1 \leq i \leq j.$$

Тогда существует такая программа $DN_{j+m_{i1}}$, вычисляющая совокупность термов $\langle t'_{j+m_{i1}} \rangle$, что

$$(3.3) \quad h(t'_{j+m_{i1}-i}) = h(t'_{j+m_{i1}-i-1}) + 1, 0 \leq i \leq m_{i1} + j - 2,$$

$$(3.4) \quad cc(\langle t'_{j+m_{i1}} \rangle) = cc(\langle t_j \rangle) + m_{i1},$$

$$(3.5) \quad h(t'_i) = h(t_i), 1 \leq i \leq j - K_1 - 1,$$

$$h(t'_{i+m_{i1}-g}) = h(t_i), 1 \leq g \leq i1-1, j-K_{i1-g}+m_{i1-g} \leq i \leq j-K_{i1-g+1}-1+m_{i1-g},$$

$$h(t'_{i+m_{i1}}) = h(t_i), j - K_{i1} \leq i \leq j,$$

$$(3.6) \quad h(t'_{j-K_{u-1}+i+m_{u-1}}) = h(t_{j-K_{u-1}}) + i, 1 \leq u \leq i1, 1 \leq i \leq P_u,$$

$$(3.7) \quad \text{все термы } t'_i \text{ независимы, } 1 \leq i \leq j + m_{i1}.$$

Доказательство. Индукция по количеству разрывов $i1$. Покажем, что

$$DN_{j+m_{i1}} = \{DN_{j+m_{i1}-1}; \{v_1 \leftarrow v_{j+m_{i1}-1} \cdot v_{j+m_{i1}-1};$$

$$[S_{(j-K_{i1}-1)+i}; v_1 \leftarrow v_{(j-K_{i1}-1)+i} \cdot v_{(j-K_{i1}-1)+i}]_{l=1}^{P_{i1}}\}; DN_j^{m_{i1}}\}$$

является искомой программой. Здесь $DN_j^{m_{i1}}$ получена из DN_j следующей заменой имен переменных:

- имена переменных v_i такие, что $1 \leq i \leq j - K_1 - 1$, не меняем,
- имена переменных v_i такие, что $j - K_1 \leq i \leq j - K_2 - 1$, меняем соответственно на имена v_i такие, что $j - K_1 + m_1 \leq i \leq j - K_2 - 1 + m_1$, и так далее, наконец,
- имена переменных v_i такие, что $j - K_{i1} \leq i \leq j$, меняем, соответственно, на такие v_i , что $j - K_{i1} + m_{i1} \leq i \leq j + m_{i1}$.

Понятно, что высоты и сложности построенных программой $DN_j^{m_{i1}}$ термов совпадают, соответственно, с высотами и сложностями термов, построенных программой DN_j .

Базис индукции. $i1 = 1$. Тогда $m_{i1} = m_1 = P_1$, то есть существует единственный разрыв глубиной P_1 с номером K_1 , $DN_{j+m_0} = DN_j$,

$$DN_{j+P_1} = \{DN_j; \{v_1 \leftarrow v_j \cdot v_j;$$

$$[S_{(j-K_1-1)+i}; v_1 \leftarrow v_{(j-K_1-1)+i} \cdot v_{(j-K_1-1)+i}]_{l=1}^{P_1}\}; DN_{j+P_1}\}.$$

Ясно, что программа DN_{j+P_1} есть программа $\{R_j; DG_p; R_j^p\}$ из Леммы 2, при $i1 = 1$ (3.1) - (3.2.1) эквивалентны (2.1) - (2.2.1), и базис обоснован по Лемме 2.

Индуктивный переход. Пусть Лемма 3 верна при всех значениях, меньших $i1 + 1$. Покажем, что Лемма 3 справедлива при $i1 + 1$.

Пусть, как это следует из (3.1) - (3.2.1), в совокупности термов $\langle t_j \rangle$, вычисляемых программой RN_j , содержится $i1 + 1$ разрыв с глубинами от P_1 до P_{i1+1} :

$$(1) \quad \langle t_j \rangle = \{t_j, \dots, t_{j-K_{i1+1}}, t_{j-K_{i1+1}-1}, \dots, t_{j-K_{i1}}, t_{j-K_{i1}-1}, \dots, t_{j-K_1}, t_{j-K_1-1}, \dots, t_1\}.$$

Покажем, что программа

$$DN_{j+m_{i1+1}} = \{DN_{j+m_{i1}}; \{v_1 \leftarrow v_{j+m_{i1}} \cdot v_{j+m_{i1}};$$

$$[S_{(j-K_{i1+1}-1)+i}; v_1 \leftarrow v_{(j-K_{i1+1}-1)+i} \cdot v_{(j-K_{i1+1}-1)+i}]_{l=1}^{P_{i1+1}}\}; DN_j^{m_{i1+1}}\}$$

ликвидирует $i1 + 1$ разрыв с глубинами от P_1 до P_{i1+1} и вычисляет удовлетворяющие (3.3) - (3.7) термы

$$(2) \langle t'_{j+m_{i1+1}} \rangle = \{t'_{j+m_{i1+1}}, \dots, t'_{j-K_{i1+1}+m_{i1+1}}, t'_{j-K_{i1+1}-1+m_{i1}}, \dots, t'_{j-K_{i1}+m_{i1}}, t'_{j-K_{i1}-1+m_{i1-1}}, \dots, t'_{j-K_1+m_1}, t'_{j-K_1-1}, \dots, t'_1\}.$$

Работа программы $DN_{j+m_{i1+1}}$ начинается с выполнения $DN_{j+m_{i1}}$, которая по индуктивному предположению ликвидирует $i1$ разрыв с глубинами от P_1 до P_{i1} , используя m_{i1} переменных дополнительной памяти и создавая совокупность термов

$$(3) \langle t^1_{j+m_{i1}} \rangle = \{t^1_{j+m_{i1}}, \dots, t^1_{j-K_{i1}+m_{i1}}, t^1_{j-K_{i1}-1+m_{i1-1}}, \dots, t^1_{j-K_1-1}, \dots, t^1_1\}.$$

Ясно, что здесь, например, $t^1_{j+m_{i1}}, \dots, t^1_{j-K_{i1}+m_{i1}}$ есть $t_j, \dots, t_{j-K_{i1}}$ с точностью до значения термина в v_1 , используемого при построении указанных термов.

Остающийся разрыв с номером K_{i1+1} ликвидирует, используя P_{i1+1} дополнительных переменных, последовательность операторов

(4) $\{S_{(j-K_{i1+1}-1)+l}; v_1 \leftarrow v_{(j-K_{i1+1}-1)+l} \cdot v_{(j-K_{i1+1}-1)+l}\}_{l=1}^{P_{i1+1}}$; создавая совокупность термов

$$(5) \{t^2_{j-K_{i1+1}-1+m_{i1+1}}, \dots, t^2_{j-K_{i1+1}-1+m_{i1}}\}.$$

(6) $DN_j^{m_{i1+1}}$ вычисляет в переменных, не использовавшихся для ликвидации разрывов, совокупность термов $\langle t^3_j \rangle$, соответствующих по высотам и сложностям термам из (1), используя при этом следующую замену номеров переменных: переменные с номерами $j-K_1 \leq i \leq j-K_2-1$ меняются, соответственно, на

$j-K_1+P_1 \leq i \leq j-K_2-1+P_1$, и так далее, наконец, переменные с номерами $j-K_{i1} \leq i \leq j-K_{i1+1}-1$ меняются, соответственно, на $j-K_{i1}+m_{i1} \leq i \leq j-K_{i1+1}-1+m_{i1}$, и переменные с номерами $j-K_{i1+1} \leq i \leq j$ меняются, соответственно, на $j-K_{i1+1}+m_{i1+1} \leq i \leq j+m_{i1+1}$.

Таким образом, вычисляемая программой $DN_{j+m_{i1+1}}$ совокупность термов

$\langle t'_{j+m_{i1+1}} \rangle$ очевидно удовлетворяет (3.3)-(3.6). (3.7) следует из того, что формируемые совокупности термов (3) и (5) перед началом работы (4) и (6) в переменной v_1 получают новое, не существовавшее ранее до этого момента, значение, соответствующее дереву высоты 0. Лемма доказана. \square

6. Использование независимого надтерма

Лемма 4. Пусть $i^*, j, k \in \mathbb{N}, i^* > 0$, программа $P_j : P_j(\sigma)(v_i) = t_i, 1 \leq i \leq j$, вычисляет совокупность термов $\langle t_j \rangle$:

$$(4.1) h(t_{i+1}) = h(t_i) + 1, i^* \leq i \leq j-1,$$

$$(4.2) \exists \langle t_j \rangle \text{ существует терм } \langle t_k \rangle : \text{надтерм } \overline{t_k} \text{ независим, } \overline{h(t_k)} = h(t_{i^*}),$$

$$(4.3) \text{Все } t_i \text{ независимы, } i^* \leq i \leq j.$$

Тогда существует программа $\overline{P1}_j$, вычисляющая $t'_j(\overline{P1}_j(\sigma)(v_j) = t'_j)$:

$$(4.4) h(t'_j) = h(t_j) + 1,$$

$$(4.5) cc(t'_j) = cc(\langle t_j \rangle),$$

$$(4.6) 0(t'_j) - \text{правильный слой.}$$

Доказательство. Легко проверить, что программа

$$\overline{P1}_j = \{v_{i^*} \leftarrow v_k \cdot v_{i^*}; [v_{i^*} \leftarrow v_i \cdot v_{i^*}]_{i=i^*+1}^j; v_j \leftarrow v_{i^*}\}$$

вычисляет терм t'_j , удовлетворяющий (4.4) - (4.6). Лемма доказана. \square

7. Построение терма высоты h в $h-m$ переменных для любого натурального m

Теорема 1. Пусть $m, j \in \mathbb{N}$. Для любого m найдется такая программа $P_j : P_j(\sigma)(v_i) = t_i$, вычисляющая совокупность термов $\langle t_j \rangle$, что

$$(1.1) \text{cc}(t_j) = h(t_j) - m.$$

Доказательство. Индукция по m . Базис индукции. $m = 0$. Легко видеть, что для терма t_2 (см. Рис. 1) теорема справедлива (пример, впервые приведенный М.А. Тайцлиным).

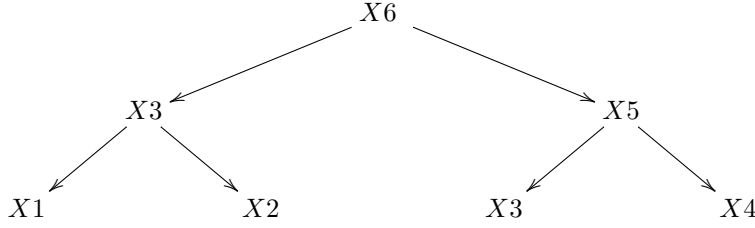


Рис. 1: Терм t_2 высоты 2, построенный с использованием 2 переменных

Индуктивный переход. Пусть теорема 1 верна при $m = k$, то есть существует программа $P_{j-1}(P_{j-1}(\sigma)(v_i) = t_i, 1 \leq i \leq j-1)$, вычисляющая совокупность термов $\langle t_{j-1} \rangle$:

$$(0.0) \text{cc}(t_{j-1}) = h(t_{j-1}) - k.$$

Итак, P_{j-1} вычисляет термы в $(j-1)$ ячейке. Отсюда, из индукционного предположения и определения вычислительной сложности получим, что $j-1 = h(t_{j-1}) - k$, то есть

$$(0.1) j = h(t_{j-1}) - k + 1.$$

Докажем, что теорема верна при $m = k + 1$. Выполнив программу

$RP_j = \{P_{j-1}; v_j \leftarrow v_{j-1}; P_{j-1}; v_j \leftarrow v_j \cdot v_{j-1}; P_{j-1}\}$ ($RP_j(\sigma)(v_i) = t_i, 1 \leq i \leq j$), получим совокупность термов $\langle t_{j-1} \rangle$:

$$(0.2) \text{ все } t_i, 1 \leq i \leq j, \text{ независимы по построению } RP_j,$$

$$(0.3) h(t_j) = h(t_{j-1}) + 1,$$

$$(0.4) \text{cc}(t_j) = \text{cc}(t_{j-1}) + 1.$$

Из (0.3), (0.4) и (0.0) получим

$$(0.5) \text{cc}(t_j) = h(t_j) - k.$$

Программа RP_j , вычисляющая $\langle t_j \rangle$, удовлетворяет условиям леммы 1, следовательно, существует программа $S_{j+h(t_j)}(S_{j+h(t_j)}(\sigma)(v_i) = t'_i, 1 \leq i \leq j + h(t_j))$, вычисляющая совокупность термов $\langle t'_{j+h(t_j)} \rangle$, для которых выполнены (1.1) - (1.6) Леммы 1 и (1.1.1) - (1.1.2) Следствия 1.1. Но тогда совокупность термов $\langle t'_{j+h(t_j)} \rangle$ удовлетворяют условиям Леммы 4 (программой P_j Леммы 4 является программа $S_{j+h(t_j)}$), и существует такая программа $\overline{P1}_{j+h(t_j)}(\overline{P1}_{j+h(t_j)}(\sigma)(v_i) = t''_i, 1 \leq i \leq j + h(t_j))$, что для совокупности термов $\langle t''_{j+h(t_j)} \rangle$, которую она вычисляет, верно (4.4) - (4.6). Из (1.2) Леммы 1 и (0.5) получим

$$(1) \text{cc}(t'_{j+h(t_j)}) = h(t_j) - k + h(t_j). \text{ Из (1) и (1.1) Леммы 1:}$$

(2) $cc(t'_{j+h(t_j)}) = h(t'_{j+h(t_j)}) - k$. Из (2) и (4.4), (4.5) Леммы 4, учитывая, что $h(t'_{j+h(t_j)}) = h(t''_{j+h(t_j)}) - 1$, получим

(3) $cc(t''_{j+h(t_j)}) = h(t''_{j+h(t_j)}) - (k+1)$. Индукционный шаг завершен. Теорема 1 доказана. \square

Заключение

Алгоритм, существующий по лемме 1, позволяет, выделив дополнительную память, построить независимый надтерм любой необходимой высоты. Если использовать только $h - m$ переменных, то потребуются ликвидировать разрывы глубиной более 1 в лесу полученных термов с разностью высот в единицу, построенных по Лемме 1. Алгоритм, существующий по Лемме 2, используя дополнительную память, обеспечивает ликвидацию разрыва любой фиксированной глубины. Таких разрывов в лесу полученных термов может быть несколько, и существующий по Лемме 3 алгоритм все эти разрывы ликвидирует, вновь затрачивая дополнительную память. Приводится эффективный алгоритм построения термов, таких, что листья соответствующих им полных бинарных деревьев помечены попарно различными подтермами, и в Теореме 1 доказывается, что для любого натурального m можно построить такой терм высоты h в $h - m$ переменных.

Список литературы

- [1] Tiuryn J. Implicit definability of finite binary trees by sets of equations // *Logic and Machines: Decision Problems and Complexity*. Springer Berlin Heidelberg, 1984. Pp. 320–332. doi:10.1007/3-540-13331-3_49
- [2] Kfoury A.I. The pebble game and logic of programmes // *Harrey Friedmans Research on the Foundations of Mathematics*. North-Holland, 1985. Pp. 317–329.
- [3] Мусикаев И.Х., Тайцлин М.А. О динамических теориях свободных алгебр // *Математический сборник*. 1989. Т. 180, № 3. С. 307–321.

Библиографическая ссылка

Дадеркин Д.О. Эффективные алгоритмы построения термов минимальной вычислительной сложности // *Вестник ТвГУ. Серия: Прикладная математика*. 2016. № 4. С. 21–33.

Сведения об авторах

1. Дадеркин Дмитрий Ольгердович

доцент кафедры информатики Тверского государственного университета.

Россия, 170100, г. Тверь, ул. Желябова, д. 33, ТвГУ.

EFFECTIVE ALGORITHM FOR CONSTRUCTION OF TERMS OF MINIMUM COMPUTATIONAL COMPLEXITY

Daderkin Dmitriy Olgerdovich

Associate Professor at Computer Science department
Russia, 170100, Tver, 33 Zhelyabova str., TSU.

Received 07.11.2016, revised 03.12.2016.

The problem of construction of terms of the minimum computational complexity by the program which contains only assignments and a finite number of variables, and operating on the free one-generated groupoid with the signature consisting of a binary operation symbol and one generating element is considered. Such terms are considered that leaves of the corresponding full binary trees are labeled by pairwise different subterms. The effective algorithm of such construction is given and it is proved that for any natural m it is possible to construct a term of height h with $h - m$ variables.

Keywords: binary tree, computational complexity, the one-generated groupoid, variable, assignment, term, effective algorithm.

Bibliographic citation

Daderkin D.O. Effective algorithm for construction of terms of minimum computational complexity. *Vestnik TverGU. Seriya: Prikladnaya Matematika* [Herald of Tver State University. Series: Applied Mathematics], 2016, no. 4, pp. 21–33. (in Russian)

References

- [1] Tiuryn J. Implicit definability of finite binary trees by sets of equations. *Logic and Machines: Decision Problems and Complexity*. Springer Berlin Heidelberg, 1984. Pp. 320–332. doi:10.1007/3-540-13331-3_49
- [2] Kfoury A.I. The pebble game and logic of programmes. *Harrey Friedmans Research on the Foundations of Mathematics*. North-Holland, 1985. Pp. 317–329.
- [3] Musikaev I.Kh., Taitslin M.A. About dynamic theories of free algebras. *Matematicheskii sbornik*, 1989, vol. 180(3), pp. 307–321 (in Russian)