УДК 510.58 681.142.2

ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ РАСПОЗНАВАНИЯ ТРАССИРУЕМОСТИ КОНФИГУРАЦИЙ

Костенко К. И.[1]

COMPUTATIONAL COMPLEXITY OF TRACEABILITY RECOGNITION CONFIGURATIONS

Kostenko K. I.

This paper presents algorithms testing possibility of c- and p-tracing between configurations of abstract knowledge spaces at n^2 time, measured by quantity of vertex marking comparisons and structural representations.

Keywords: semantic structure, tracing of structures, algorithm complexity

Введение

Пространства знаний — это интеллектуальные системы, отражающие целостные семейства знаний произвольных областей деятельности и технологии работы с ними. Абстрактные пространства знаний являются формализмами пространств знаний, предназначенными для исследования фундаментальных свойств формализованных знаний и операций над ними точными методами.

Элементы таких пространств рассматриваются в качестве абстрактных знаний и называются конфигурациями. Они представляются нагруженными бинарными деревьями, в которых висячим вершинам сопоставляются элементарные (неделимые) конфигурации, а остальным вершинам — семантические отношения, выполняющиеся для пар конфигураций, представленных левым и правым поддеревом таких вершин.

Трассирование конфигураций определяет зависимость между ними, связанную с существованием таких изотонных отображений структуры одной конфигурации в структуру другой, что разметки сопоставляемых друг другу вершин оказываются сравнимыми [1]. Практическая применимость трассирований конфигураций связана с построением алгоритмов его распознавания, имеющих приемлемую вычислительную сложность.

1. Основные определения

Обозначим как $\mathbf{M}$ бесконечное вычислимое множество, элементы которого называются конфигурациями, содержащее пустую конфигурацию Λ.

Пусть $\mathbf{R}$ — вычислимое множество разрешимых бинарных отношений на $\mathbf{M}$, на котором определено разрешимое отношение вложения отношений ρ_1.

Отношения $\mathbf{T} = \mathbf{M} \times \mathbf{M}$ и $\mathbf{E} = \varnothing$ являются элементами $\mathbf{R}$. $\mathbf{E}$ связывает любые конфигурации, представляя отсутствие зависимости между ними.

Определение. Разложением (разложением конфигураций) называется всюду определенное вычислимое отображение $\varepsilon : \mathbf{M} \to \mathbf{M} \times \mathbf{M}$, для которого

$$\varepsilon(\Lambda) = (\Lambda, \Lambda);$$

$$\forall\, z_1, z_2 \in \mathbf{M}\ \exists\, z \in \mathbf{M}\ \big(\varepsilon(z) = (z_1, z_2)\big).$$

Конфигурация z называется элементарной в разложении ε если $\varepsilon(\Lambda) = (\Lambda, \Lambda)$. Обозначим как $\mathbf{M}_0^\varepsilon$ ($\mathbf{M}_1^\varepsilon$) множество элементарных (неэлементарных) конфигураций в разложении ε. Будем рассматривать разложения ε с бесконечными множествами $\mathbf{M}_0^\varepsilon$, на которых определены вычислимые отношения порядка ρ_0 с минимальным элементом Λ.

Пусть $\mathbf{N} = \{0, 1, \ldots\}$. Глубиной разложения ε называется отображение $d_\varepsilon : \mathbf{M} \to \mathbf{N}$,

[1]Костенко Константин Иванович, канд. физ.-мат. наук, доцент кафедры информационных технологий, начальник отдела разработки информационных систем Центра интернет Кубанского государственного университета; e-mail: kostenko@kubsu.ru

определяемое соотношениями

$$d_\varepsilon(z) = 0 \leftrightarrow z = \Lambda;$$

$d_\varepsilon(z) = \max(d_\varepsilon(z_1), d_\varepsilon(z_2)) + 1$, если $\varepsilon(z) = (z_1, z_2)$ и $z \neq \Lambda$.

Если отображение d_ε является всюду определённым, то разложение ε называется конечным.

Если $z_1, z_2 \in \mathbf{M}$, обозначим как $\eta^\varepsilon_{z_1,z_2}$ однозначную вычислимую нумерацию множества $\mathbf{M}_\varepsilon(z_1, z_2) = \{z|\varepsilon(z) = (z_1, z_2)\}$

Определение. Вычислимое отображение $\psi : \mathbf{M}^\varepsilon_1 \to \mathbf{R}$ называется семантическим связыванием для разложения ε, если

1. $\forall\ z_1, z_2 \in \mathbf{M}\ \exists\ z \in \mathbf{M}\ (z_1 \neq \Lambda \vee z_2 \neq \Lambda \to$
   $\to \varepsilon(z) = (z_1, z_2)\ \&\ \psi(z) = \mathbf{E})$;

2. $\forall\ z \in \mathbf{M}^\varepsilon_1\ (\psi(z) \neq \mathbf{E} \to \varepsilon(z) \in \psi(z))$;

3. $\eta^\varepsilon_{z_1,z_2}$ порождает однозначную вычислимую нумерацию множества $\{r \mid \exists z \in \mathbf{M}_\varepsilon(z_1, z_2)\ (r = \psi(z))\}$, если $z_1 \neq \Lambda \vee z_2 \neq \Lambda$, определяемую соотношением

$$\nu(n) = r \Leftrightarrow \psi(\eta^\varepsilon_{z_1,z_2}(n)) = r.$$

Определение. Пространством конфигураций называется всякая пара $\mathbf{M} = (M, d)$, где $\mathbf{M}$ — бесконечное разрешимое множество конфигураций, содержащее Λ, а $d = (\varepsilon, \psi)$ — декомпозиция элементов $\mathbf{M}$.

Здесь $\varepsilon : \mathbf{M} \to \mathbf{M} \times \mathbf{M}$ — разложение, а $\psi : M^\varepsilon_1 \to \mathbf{R}$ — семантическое связывание для ε.

2. Трассирования конфигураций

Декомпозиция конфигураций $d = (\varepsilon, \psi)$ со всюду определенной функцией глубины разложения d_ε порождает полные структурные представления (ПСП) элементов $\mathbf{M}$ в виде конечных нагруженных бинарных деревьев. Корню ПСП $z \in \mathbf{M}^\varepsilon_1$ сопоставляется $\psi(z) \in \mathbf{R}$, а его левое и правое поддеревья образуют ПСП конфигураций из $\varepsilon(z)$. Листьям ПСП конфигурации z соответствуют элементы $\mathbf{M}^\varepsilon_0$.

Вершины ПСП именуются двоичными словами, так что пустому слову λ соответствует корень. Если слову α сопоставлена вершина ν, то словам $\alpha 0$ и $\alpha 1$ соответствуют левый и правый потомки ν. Обозначим как $\mathbf{D}(z)$ ($\mathbf{O}(z)$) множество вершин (листьев) ПСП $z \in \mathbf{M}$. Если $z \in \mathbf{M}$ и $\alpha \in \mathbf{D}(z)$, то $(z)^\varepsilon_\alpha$ обозначает подконфигурацию конфигурации z, определяемую поддеревом ПСП z с корнем α. Разметку $\alpha \in D(z)$ обозначим через $[z]^\varepsilon_\alpha$.

Определим сравнения трассирования конфигураций, основанные на изотонных отображениях их структур, со сравнимыми в ρ_0 и ρ_1 разметками соответствующих вершин.

Пусть $\mathbf{I}$ — множество конечных двоичных последовательностей, содержащее пустую последовательность λ, и $z_1, z_2 \in \mathbf{M}$.

Определение Изотонное отображение $\xi : \mathbf{I} \to \mathbf{I}$ называется трассированием $z_1 \in \mathbf{M}$ в $z_2 \in \mathbf{M}$, если

$$\xi(\mathbf{D}(z_1)) \subseteq \mathbf{D}(z_2)\ \&\ \forall\alpha \in \mathbf{D}(z_1)$$
$$(\alpha \in \mathbf{D}(z_1)\backslash\mathbf{O}(z_1) \leftrightarrow \xi(\alpha) \in \mathbf{D}(z_2)\backslash\mathbf{O}(z_{12}));$$

$$\forall\alpha, \alpha\sigma \in \mathbf{D}(z_{11}), \sigma \in \{0, 1\}$$
$$\exists\,\beta, \gamma \in \mathbf{I}((\xi(\alpha) \subset \xi(\alpha\sigma) \to \xi(\alpha\sigma) = \xi(\alpha)\beta\sigma\gamma).$$

Специальными классами трассирований являются o-трассирования ($\beta = \lambda$), p-трассирования ($\beta = \lambda$ и ξ-инъективное на $D(z_1)$), и c-трассирования ($\beta = \gamma = \lambda$) [1].

Конфигурация z_1 I-трассируется ($I \in \{o, p, c\}$) в z_2 (обозначается как $z_1 \leqslant_I z_2$), если существует такое I-трассирование ξ конфигурации z_1 в z_2, что

$$\forall\alpha \in \mathbf{O}(z_1)((z_1)_\alpha\ \rho_0\ (z_2)_{\xi(\alpha)});$$

$$\forall\alpha \in \mathbf{D}(z_1)\backslash\mathbf{O}(z_1)([z_1]_\alpha\ \rho_1\ [z_2]_{\xi(\alpha)}).$$

Пусть $z_1, z_2 \in \mathbf{M}$. Рассмотрим задачу оценки вычислительной сложности алгоритмов распознающих c- и p-трассируемость конфигурации z_1 в z_2.

Будем оценивать вычислительную сложность алгоритма проверки трассируемости конфигураций с помощью количества сравнений разметок вершин ПСП конфигураций в отношениях ρ_0 и ρ_1.

3. Алгоритм распознавания c-трассируемости конфигураций

Приведём алгоритм распознавания c-трассируемости произвольных конфигураций z_1 и z_2, основанный на обходе дерева ПСП z_1 в глубину сверху вниз и слева направо, имеющий полиномиальную сложность [4]. Такой алгоритм проверяет возможность определения отображения c-трассирования ξ для z_1 в z_2, для которого

$\lambda \in \mathbf{D}(z_1)$ в качестве значения $\xi(\lambda)$ перебираются элементы $\mathbf{D}(z_2)$ также в порядке прохождения этого дерева в глубину сверху вниз. В каждом случае проверяется возможность доопределения подходящего отображения ξ, для которого выполнены условия c-трассирования z_1 в z_2.

При этом при определении значения $\xi(\alpha)$ для очередной вершины $\alpha \in \mathbf{D}(z_1)$ может потребоваться переопределение уже назначенных значений ξ для вершин из $\mathbf{D}(z_1)$.

Определяемый ниже алгоритм основывается на следующих свойствах c-трассирований:

1. Если ξ — это c-трассирование $z_1 \in \mathbf{M}$ в $z_2 \in \mathbf{M}$, то самая левая ветвь в $\mathbf{D}(z_1)$ отображается с помощью ξ на самую левую ветвь в $\mathbf{D}(z_2)_{\xi(\alpha)}$.

2. Множества вершин из $\mathbf{D}(z_1)$, отображаемых ξ в конкретные вершины из $\mathbf{D}(z_2) \cap \mathbf{I}_{\xi(\lambda)}$, составляют вершины одного или нескольких бинарных деревьев.

3. Если ξ — это c-трассирование $z_1 \in \mathbf{M}$ в $z_2 \in \mathbf{M}$ и $\xi(\beta) = \alpha$, где $\beta = \delta_1 \dots \delta_p$, $\alpha = \sigma_1 \dots \sigma_k$, то последовательность слов, являющихся началами β, сюрьективно сопоставляется последовательности слов, являющихся всеми началами α.

Алгоритм нахождения отображения использует разбиение применяемой схемы обхода ПСП z_1 в глубину на этапы непрерывного прохождения в глубину по ветвям, образуемым левыми потомками последовательно проходимых вершин в ПСП вплоть до висячих вершин. Такие прохождения реализуются в начале обхода, а также всякий раз после перехода к правым потомкам ранее пройденных вершин.

Пусть $\gamma \in \mathbf{D}(z_2)$ и $\xi(\lambda) = \gamma$. Без ограничения общности будем считать, что конфигурации z_1 и z_2 не являются элементарными. Процесс построения отображения c-трассирования составляется из двух этапов.

Этап 1 (Определение ξ для крайней левой ветви в $\mathbf{D}(z_1)$)

Пусть $W_2 = \alpha^0, \alpha^1, \dots, \alpha^k$ последовательность вершин, образующая самую левую ветвь в $\mathbf{D}(z_2)$, ведущую из γ в висячую вершину. Обозначим как $W_1 = \beta^0, \beta^1, \dots, \beta^d$ аналогичную последовательность вершин в $D(z_1)$, где $\beta^0 = \lambda$.

Выполним подбор подходящих значений ξ для элементов W_1. По свойствам c-трассирований определяемое алгоритмом отображение ξ должно быть сюрьекцией W_1 на W_2 так, что висячая вершина в W_1 сопоставляется висячей вершине из W_2.

a) Каждому $a^i \in W_2$ поставим в соответствие последовательно конструируемый список L^i — вершин из W_2, отображаемых ξ в a^i, который сначала является пустым.

b) Проверим выполнимость отношения $[z_1]_\lambda \rho_1 [z_2]_\gamma$, если $\lambda \in \mathbf{D}(z_1) \backslash \mathbf{O}(z_1)$, или $[z_1]_\lambda \rho_0 [z_2]_\gamma$ — в противном случае.

Если это не так, то c-трассирование ξ конфигурации z_1 в z_2, для которого $\xi(\lambda) = \gamma$, не существует. Изменим значение $\xi(\lambda)$ на следующий проходимый в глубину элемент $\mathbf{D}(z_2)$ и повторим предыдущее действие, а если такого элемента нет, то с-трассирование z_1 в z_2 невозможно.

Пусть отношение $[z_1]_\lambda \ \rho_1 \ [z_2]_\gamma$ (или $[z_1]_\lambda \ \rho_0 \ [z_2]_\gamma$) выполнено.

Образуем вспомогательный список L_1, в который включим элементы W_1, выписанные в обратном порядке. Распределим элементы L_1 по спискам $L^0, \dots, L^k$, соответствующим вершинам $\alpha^0, \alpha^1, \dots, \alpha^k$, так, чтобы разметка соответствующих им висячих и внутренних вершин в $\mathbf{D}(z_1)$ находилась в отношениях ρ_0 и ρ_1 с соответствующими им вершинами из $\mathbf{D}(z_2)$.

Для этого переместим в начало списка L^0 слева направо все такие идущие подряд элементы конца списка L_1, содержащего γ, для которых выполняется условие $\forall \ \beta \in L_0([z_1]_\beta \ \rho_1 \ [z_2]_{\alpha^0})$, если $\alpha^0 \in \mathbf{D}(z_1) \backslash \mathbf{O}(z_1)$. Если же $\alpha^0 \in \mathbf{O}(z_1)$ и $[z_1]_\lambda \ \rho_0 \ [z_2]_\gamma$, то переместим в L^0 только значение γ. Для перемещаемых вершин положим $\xi(\beta) = \gamma$. Определим значение $p = 0$.

c) Пусть η — последний элемент в L_1. Проверим условие $[z_1]_\eta \, \rho_1 \, [z_2]_{\alpha^{p+1}}$, если $\eta \in \mathbf{D}(z_1) \backslash \mathbf{O}(z_1)$ и $\alpha^{p+1} \in \mathbf{D}(z_2) \backslash \mathbf{O}(z_2)$, или $[z_1]_\eta \, \rho_0 \, [z_2]_{\alpha^{p+1}}$, если $\eta \in \mathbf{O}(z_1)$ и $\alpha^{p+1} \in \mathbf{O}(z_2)$.

c1) Если условие $[z_1]_\eta \, \rho_1 \, [z_2]_{\alpha^{p+1}}$ ($[z_1]_\eta \, \rho_0 \, [z_2]_{\alpha^{p+1}}$) выполнено, переместим в начало списка L^{p+1} слева направо все такие подряд идущие элементы конца списка L_1, содержащего η, для которых выполняется условие $\forall \ \beta \in L_1([z_1]_\beta \, \rho_1 \, [z_2]_{\alpha^{p+1}})$, если $\alpha \in \mathbf{D}(z_1) \backslash \mathbf{O}(z_1)$. Если же $\eta \in \mathbf{O}(z_1)$ и $[z_1]_\eta \, \rho_0 \, [z_2]_{\alpha^{p+1}}$, переместим в L^{p+1} только значение η. Для перемещаемых в L^{p+1} элементов положим значение ξ равным α^{p+1}. Увеличим значение p на 1. Повторим действие *c)*.

c2) Пусть условие $[z_1]_\eta\,\rho_1\,[z_2]_{\alpha^{p+1}}$ ($[z_1]_\eta\,\rho_0\,[z_2]_{\alpha^{p+1}}$) не выполнено. Тогда в последовательно просматриваемых списках $L^p,\ldots,L^0$ выполним поиск такого первого элемента $\omega\in L^t$, для которого $[z_1]_\omega\,\rho_1\,[z_2]_{\alpha^{t+1}}$ и ω не является последним элементом в списке L^t

Если значение ω не найдено, то c-трассирование z_1 в z_2, для которого $\xi(\lambda)=\gamma$, не существует. В этом случае изменим значение $\xi(\lambda)$ на следующий проходимый в глубину элемент $\mathbf{D}(z_2)$ и выполним построение ξ заново. Если же такого элемента нет, то c-трассирование z_1 в z_2 невозможно.

Пусть подходящий $\omega\in L^t$ найден. Добавим все просмотренные элементы, включая ω к списку L_1 справа, сохранив информацию о вершинах ПСП z_1, которым они были сопоставлены. Изменим значение p на t. После этого выполним действия шага *c)*. Если в процессе перемещения элементов L_1 в списки $L^0,\ldots,L^k$ встретится вершина η, которая перемещается в список L^i, назначенный ей ранее, то выполнение шага c продолжается с действия *c2)*.

Данный этап завершается за конечное число шагов, поскольку при каждом повторном выполнении шага *c)* этап либо завершается, либо происходит перемещение хотя бы одного элемента из списков L^0, $L^1,\ldots,L^k$ в список с бо́льшим индексом.

Если процесс перемещения элементов всех L_1 по спискам $L^0,\ldots,L^k$ завершается успешно, то для его результата выполняется условие

$$\forall\ i=1,\ldots,k\ \forall\ \alpha\in L^i(\alpha\in D(z_1)\backslash O(z_1)\to \\ \to[z_1]_\alpha\,\rho_1\,[z_1]_{\alpha^i}\&$$

$$\alpha\in O(z_1)\to[z_1]_\alpha\,\rho_0\,[z_1]_{\alpha^i}).$$

Если же выполнение этапа завершается неудачей, то это означает c-трассирование ξ, конфигурации z_1 в z_2, для которого $\xi(\lambda)=\gamma$, не существует.

Данный этап завершается за конечное число шагов, поскольку при каждом повторном выполнении действия *c) этап 1* либо завершается, либо происходит перемещение хотя бы одного элемента из списков L^0, $L^1,\ldots,L^k$ в списки с большими индексами.

Отметим свойства приведенной части алгоритма, необходимые для обоснования его правильности и оценки сложности

I. Задаваемые на данном этапе значения отображения ξ реализуют схему «жадного» алгоритма, в которой значения $\xi(\gamma)$, $\gamma\in\mathbf{D}(z_1)$, определяются так, чтобы быть максимально близким к значению $\xi(\lambda)$.

II. Всякое повторное выполнение шага *c)* приводит к такому перераспределению элементов рассматриваемой части W_1 по спискам, соответствующим элементам W_2, при котором хотя бы один элемент W_1 перемещается в список с большим номером.

III. Если этап завершается успешно, то определённый фрагмент отображения ξ соответствует трассированию сжатия для полностью пройденной в глубину самой левой ветви в $\mathbf{D}(z_1)$ в подходящую ветвь $\mathbf{D}(z_2)$.

IV. Неудача в определении ξ в рассматриваемом случае означает, что последовательность W_1 нельзя сжать в последовательность W_2, так чтобы разметки соответствующих вершин находились в отношениях ρ_0 и ρ_1.

Этап 2 (Определение отображения ξ для правых потомков вершин $\mathbf{D}(z_1)$)

При обходе $\mathbf{D}(z_1)$ в глубину правые потомки произвольных вершин проходятся при первом возврате в эти вершины после полного прохождения левых поддеревьев таких вершин. После перехода к правому потомку некоторой вершины прохождение дерева $\mathbf{D}(z_1)$ продолжается из этой вершины по выходящей из неё самой левой ветви дерева с использованием схемы *этапа 1*. При этом отображение ξ доопределяется для заданной ветви таким образом, чтобы оставаться c-трассированием пройденной части $\mathbf{D}(z_1)$ в некоторый фрагмент $\mathbf{D}(z_2)$.

Пусть для вершины $\beta\in\mathbf{D}(z_1)$ реализован обход её левого поддерева так, что значения отображения ξ уже определены для всех вершин множества $\mathbf{D}(z_1)\cap\mathbf{I}_{\beta 0}$. Если $\xi(\beta)=\alpha$, обозначим как $W_1=\beta_0,\beta_1,\ldots,\beta_p$ ($W_2=\alpha_0,\alpha_1,\ldots,\alpha_k$) — путь в $\mathbf{D}(z_1)(\mathbf{D}(z_2))$ из λ в β (из λ в α). Обозначим как $L^0,L^1,\ldots,L^k$ разбиение W_1 на списки, составленные из элементов с одинаковыми значениями отображения ξ.

Для значения ξ в очередной вершине $\beta 1\in\mathbf{D}(z_1)$ выполнено условие: $\xi(\beta 1)=\xi(\beta)\vee\xi(\beta)=\xi(\beta)1$. Тогда построение ξ сначала реализуется через поиск подходящего определения ξ для вершин левой ветви в $\mathbf{D}(z_1)$, выходящей из $\beta 1$ для условия $\xi(\beta 1)=\xi(\beta)$. Если это невозможно, то определение ξ отыскивается для условия $\xi(\beta 1)=\xi(\beta)1$.

Если же оба процесса завершаются неудачей, выполняется переопределение ξ для одной из вершин $\mathbf{D}(z_1)$, предшествующих β, после чего продолжается построение ξ, начиная с этой вершины.

Приведём точное описание процесса определения ξ для данного этапа.

Проверим выполнимость условия $[z_1]_{\beta 1}\rho_1[z_2]_{\xi(\beta)}$, если $[z_1]_{\beta 1} \in \mathbf{D}(z_1)\backslash\mathbf{O}(z_1)$, или условия $[z_1]_{\beta 1}\rho_1[z_2]_{\xi(\beta)}$ — в противном случае.

Если условие выполнено, положим $\xi(\beta 1) = \alpha$.

Продолжим определение ξ для вершин левой ветви дерева $(\mathbf{D}(z_1))_{\beta 1}$, используя метод перемещения элементов списков из *этапа 1.* Результатом выполнения данного этапа является либо отображение c-трассирования ξ будет определено для вершин левой ветви в $\mathbf{D}(z_1)$, начинающейся в $\beta 1$, вплоть до вершины $\beta 1(0)^t \in \mathbf{O}(z_1)$, либо будет установлена невозможность такого определения.

В этом случае определение ξ продолжается для следующей проходимой в глубину вершины $\mathbf{D}(z_1)$, являющейся правым потомком некоторой вершины ε этого дерева, аналогично случаю вершины $\beta 1 \in \mathbf{D}(z_1)$, либо завершается успешно, если такой вершины $\varepsilon 1$ нет.

Если условие $[z_1]_{\beta 1}\rho_1[z_2]_{\xi(\beta)}$ не выполняется или для $\xi(\beta 1) = \alpha$ не удаётся доопределить ξ на левую ветвь в $\mathbf{D}(z_1)$, начинающуюся из $\beta 1$, положим $\xi(\beta 1) = \alpha 1$ и проверим условие $[z_1]_{\beta 1}\rho_1[z_2]_{\xi(\beta)1}$ (или $[z_1]_{\beta 1}\rho_0[z_2]_{\xi(\beta)1}$).

Если последнее условие выполняется, то с помощью метода перемещения элементов списков выполним поиск определения ξ для элементов левой ветви в $\mathbf{D}(z_1)$, начинающейся из вершины $\beta 1$.

Если определение ξ для элементов указанной ветви в каждом из рассмотренных случаев завершается неудачей, то изменим его по приводимой ниже схеме:

Пусть $\beta = \delta_1 \dots \delta_p$. Рассмотрим отдельно случаи для разных значений элемента δ_p.

Случай 1. Для $\delta_p = 0$ положим $\beta = \beta'\beta''$, где β' — последняя группа нулей в β. Переопределим значения ξ для вершин из $\mathbf{D}(z_1)$, проходимых в глубину, начиная с β'. Пусть $W_2 = \alpha^0, \dots, \alpha^k$ — ветвь в $\mathbf{D}(z_2)$, элементы которой сопоставляются вершинам ветви W_1 в $\mathbf{D}(z_1)$, ведущей из β' в β.

Пусть $\beta' 0^m \in \mathbf{O}(z_1)$. Как и для *этапа 1*, составим списки $L^0, \dots, L^d$, содержащие элементы W_1, сопоставляемые отображением ξ вершинам пути W_2, ведущего из β' в $\beta' 0^m$. Пусть $\beta \in L^r$. Переместим в L_1 вершины списков $L^d, L^{d-1}, \dots, L^r$, вплоть до β.

Для L_1 и $L^r, \dots, L^0$ будем повторно выполнять процедуру перемещения элементов этих списков из *этапа 1* до тех пор пока отображение ξ не изменит значение для β и не будет найдена первая справа налево вершина $\beta_0 \in L_1$, для которой значение ξ останется неизменным.

a) Если данный процесс завершится успешно, то он переопределяет ξ, которое для вершин, предшествующих β_0 в прохождении $\mathbf{D}(z_1)$ в глубину, и всех вершин поддерева $\mathbf{D}(z_1) \cap \mathbf{I}_{\beta_0}$. После этого процесс определения ξ продолжается для следующей после β_0 вершины обхода $\mathbf{D}(z_1)$ в глубину.

b) Если перемещение элементов L_1 и $L^r, \dots, L^0$ завершается неудачей и $\beta' = \lambda$, то определение трассирования ξ невозможно для заданного значения $\xi(\lambda)$. Если же $\beta' = \beta''' 1$, то продолжим переопределение ξ по последней схеме, заменив в ней β на β'''.

Случай 2. Пусть $\delta_p = 1$ и $\xi(\beta) = \sigma_1 \dots \sigma_q$.

Для $\sigma_q = 0$ положим $\xi(\beta) = \sigma_1 \dots \sigma_q 1$ после чего продолжим построение ξ для вершин самой левой ветви в $\mathbf{D}(z_1)$, начинающейся из вершины β, с помощью схемы *этапа 1.*

Если же $\sigma_q = 1$, то рассмотрим распределение вершин пути в $\mathbf{D}(z_1)$, ведущего из λ в β, по непустым спискам $L^0, \dots, L^q$, сопоставленным вершинам пути в $\mathbf{D}(z_2)$, ведущего из $\xi(\lambda)$ в $\xi(\beta) = \sigma_1 \dots \sigma_q$.

Будем последовательно просматривать списки $L^q, \dots, L^0$, в порядке справа налево, до тех пор пока не найдём в них вершину $\omega = \delta_1 \dots \delta_m$, не являющуюся первой в соответствующем списке, для которой выполнено одно из условий:

1. $\delta_m = 0$ & $[z_1]_\omega \rho_1 [z_2]_{\xi(\delta_1 \dots \delta_m)0}$;
2. $\delta_m = 1$ & $[z_1]_\omega \rho_1 [z_2]_{\xi(\delta_1 \dots \delta_m)1}$.

Если имеет место первое условие, то заменим значение $\xi(\omega)$ на $\xi(\delta_1 \dots \delta_m)0$, а если второе, то положим $\xi(\omega) = \xi(\delta_1 \dots \delta_m)1$.

Продолжим построение c-трассирования ξ, начиная в вершины ω, продолжая обход $\mathbf{D}(z_1)$ в глубину.

Если вершина ω не найдена, то c-трассирование z_1 в z_2, для выбранного значения $\xi(\lambda)$ не существует. Заменим $\xi(\lambda)$ на следующий элемент $\mathbf{D}(z_2)$ в порядке обхода этого дерева в глубину и для нового значения $\xi(\lambda)$ выполним построение c-трассирования ξ. Если следующего элемента в $\mathbf{D}(z_2)$ нет, то c-трассирование z_1 в z_2 невозможно.

Отметим свойства *этапа 2*, необходимые для обоснования корректности приведенного алгоритма и оценки его вычислительной сложности.

V. Переопределение отображения ξ для вершин пути из λ в β, для которой распознаётся невозможность подходящего задания ξ для вершин $\mathbf{D}(z_1)$, проходимых после β, реализует схему возврата к ближайшей вершине пути, ведущего из λ в вершину $\mathbf{D}(z_1)$, для которого такое переопределение ξ существует.

VI. Если для вершины $\beta \in \mathbf{D}(z_1)$ при возврате или перемещении элементов списков значение $\xi(\beta)$ не изменяется, то для такой вершины и её потомков в $\mathbf{D}(z_1)$ на соответствующем шаге работы приведённого алгоритма отображение ξ не изменяется, следовательно, не требует его переопределения.

Выполнение алгоритма продолжается до тех пор, пока либо будет построено подходящее отображение c-трассирования для конфигураций z_1 и z_2, либо процесс завершится неудачей для выбора в качестве начального значения $\xi(\lambda)$ последней вершины проходимого в глубину дерева $\mathbf{D}(z_1)$.

Вычислительная сложность алгоритма распознавания c-трассирования конфигураций

В качестве размера начальных данных приведённого алгоритма выберем

$$n = \max(|D(z_1)|, |Dz_2|).$$

Количество действий, выполняемых этим алгоритмом, определяется числом перемещений элементов списков при отображении левых ветвей заданных начальных вершин из $\mathbf{D}(z_1)$ в аналогичные ветви из $\mathbf{D}(z_2)$ и обратных перемещений по $\mathbf{D}(z_1)$ и $\mathbf{D}(z_2)$ при переопределении ξ.

Поэтому вычислительную сложность приведённого алгоритма можно оценить числом обращений к элементам ПСП z_1 и $(z_2)_\alpha$ при его выполнении.

Задание начальных значений отображения ξ для отдельных элементов $\mathbf{D}(z_1)$ и всякое их изменение впоследствии выполняется, либо при перемещении элементов списков для прохождения *этапа 1*, либо при возвратах для прохождения *этапа 2*. При этом для всякого переопределения значений ξ требуется просмотр последовательностей, содержащих не более n вершин ПСП конфигураций z_1 и z_2.

При изменении значений ξ для конкретных вершин из $\mathbf{D}(z_1)$ последнее заменяется на вершины из $\mathbf{D}(z_2)$, которые проходятся в $\mathbf{D}(z_2)$ после заменяемых вершин. Поэтому при определении значения ξ для всякой вершины из $\mathbf{D}(z_1)$ переопределения этого значения осуществляются не более n раз. Если для некоторого соотношения $\beta \in \mathbf{D}(z_1)$, удаётся определить c-трассирование ξ конфигурации $(z_1)_\beta$ в $(z_2)_\alpha$ для которого $\xi(\lambda) = \lambda$, то такое трассирование можно зафиксировать и использовать всякий раз, когда переопределение ξ не продолжается для вершин $\beta \in \mathbf{D}(z_1)$, для которых $\xi(\beta) = \alpha$.

Следовательно, общее время построения отображения ξ или получения вывода о невозможности подходящего c-трассирования не превышает n^2.

4. Алгоритм распознавания p-трассирования конфигураций

Трассирования растяжения порождаются инъективными отображениями трассирования. Рассмотрим алгоритм распознавания существования таких трассирований для произвольных пар конфигураций, вычислительная сложность которого не превосходит n^2.

Для произвольных конфигураций z_1 и z_2 этот алгоритм моделирует поиск подходящих значений подбираемого отображения трассирования в процессе обхода ПСП z_1 в глубину в порядке сверху вниз. При этом на первом шаге выполняется поиск такого трассирования ξ, для которого $\xi(\lambda) = \lambda$. В последующем в качестве значений $\xi(\lambda)$ выбираются вершины ПСП z_2 в порядке обхода дерева $\mathbf{D}(z_2)$, до тех пор пока не будет подобрано подходящее трассирование. Процесс построения трассирования ξ завершается неудачей, если в качестве $\xi(\lambda)$ последовательно использовались все вершины $\mathbf{D}(z_2)$, но отображение ξ осталось не полностью определённым. При построении ξ формируется вспомогательный список L, образованный такими парами (α, β), что не существует p-трассирования z_1 в z_2, для которого $\xi(\alpha) = \beta$. Пары (α, β) включаются в L всякий раз, когда установлена невозможность p-трассирования $(z_1)_\alpha$ в $(z_2)_\beta$. Если для вершин $\alpha \in \mathbf{D}(z_1)$ и $\beta \in \mathbf{D}(z_2)$ будет найдено c-трассирование $(z_1)_\alpha$ в $(z_2)_\beta$, для которого $\xi(\alpha) = \beta$, включим пару (α, β) в список $\mathbf{R}$.

Списки L и $\mathbf{R}$ используется для того, чтобы повторно не выполнять поиск p-

трассирований в подконфигураций конфигурации z_1 в те подконфигурации z_2, для которых существование или отсутствие таких трассирований было установлено ранее.

Описание алгоритма распознавания p-трассируемости конфигураций

Рассмотрим схему одного шага алгоритма построения трассирования ξ в общем случае, когда $\xi(\lambda) = \alpha$, где $\alpha \in \mathbf{D}(z_2)$.

Проверим условие $[z_1]_\lambda\, \rho_1\, [z_2]_{\xi(\lambda)}$, если $[z_1]_\lambda \in \mathbf{D}(z_1)\backslash\mathbf{O}(z_1)$, и $[z_1]_\lambda\, \rho_0\, [z_2]_{\xi(\lambda)}$, если $[z_1]_\lambda \in \mathbf{O}(z_1)$.

Пусть такое условие не выполняется. Тогда шаг завершается неудачей. Выполним переход к следующему шагу работы алгоритма, изменив значение $\xi(\lambda)$.

Если проверяемое условие выполнено, определим множества

$$\Delta_0 = \mathbf{D}(z_2) \cap \mathbf{I}_{\xi(\lambda)0} \text{ и } \Delta_1 = \mathbf{D}(z_2) \cap \mathbf{I}_{\xi(\lambda)1}.$$

Продолжим определение значений $\xi(0)$ и $\xi(1)$, в областях $\Delta_{\alpha 0}$ и $\Delta_{\alpha 1}$ подбирая такие значения при обходе выбранных областей в глубину. Пусть на некотором шаге в качестве $\xi(0)$ ($\xi(1)$) выбрано значение $\beta \in \Delta_{\alpha 0}$. ($\beta \in \Delta_{\alpha 1}$). Определим множества $\Delta_{\beta 0}$ и $\Delta_{\beta 1}$, в которых выполним поиск значений $\xi(00)$ и $\xi(01)$ ($\xi(10)$ и $\xi(11)$) соответственно. Такой поиск выполним по той же схеме, что и для $\xi(0)$ ($\xi(1)$). Если для $\beta \in \mathbf{D}(z_1)$ отображение ξ определено для всех элементов $\mathbf{D}(z_1) \cap \mathbf{I}_\beta$, включим пару $(\beta, \xi(\beta))$ в список $\mathbf{R}$.

Если определение трассирования ξ в областях $\Delta_{\alpha 0}$ или $\Delta_{\alpha 1}$ завершается неудачей, p-трассирование ξ конфигурации z_1 и z_2, для которого $\xi(\lambda) = \alpha$, не существует. Возьмём в качестве $\xi(\lambda)$ следующую по порядку вершину области $\mathbf{D}(z_2)$. Включим (λ, α) в список L. Если такой вершины нет, то p-трассирование z_1 и z_2 невозможно.

Приведённая схема определения значений ξ применяется до тех пор, пока не потребуется определить значение ξ для $\gamma \in \mathbf{O}(z_1)$. Пусть, в этом случае $\gamma = \sigma_1, \ldots, \sigma_k$ и значение $\xi(\gamma)$ отыскивается в области $\Delta_{\omega\sigma_k}$. Среди вершин множества $\mathbf{O}(z_2) \cap \Delta_{\omega\sigma_k}$ найдём первую слева направо вершину δ, для которой $[z_1]_\gamma \rho_0 [z_2]_\delta$.

Если такая вершина найдена, положим $\xi(\gamma) = \delta$. В противном случае процесс построения ξ в области $\Delta_{\omega\sigma_k}$ завершается неудачей. Добавим (λ, α) в список L. Для этого случая выбирается следующее по порядку значение для $\xi(\sigma_1, \ldots, \sigma_{k-1})$, для которого $(\sigma_1, \ldots, \sigma_{k-1}, \xi(\sigma_1, \ldots, \sigma_{k-1})) \notin L$, поскольку p-трассирование ξ конфигураций z_1 и z_2 с $\xi(\sigma_1, \ldots, \sigma_{k-1}) = \omega$, невозможно.

Сложность алгоритма распознавания p-трассирования конфигураций

Время выполнения алгоритма определяется числом операций, которые необходимо выполнить в процессе перемещения последовательно подбираемых значений отображения трассирования для вершин из $\mathbf{D}(z_1)$. Такое перемещение для всякой вершины реализуется не более n раз, поскольку повторно не выполняется для областей, в которых установлена возможность или невозможность p-трассирований подконфигураций z_1 в подконфигурации z_2. Поэтому в процессе построения подходящего отображения трассирования ξ его значение для каждого элемента $\mathbf{D}(z_1)$ изменяется не более n раз.

Следовательно, вычислительная сложность приведённого алгоритма для конфигураций, максимум числа вершин ПСП которых равен n, не превышает n^2.

Литература

1. *Костенко К. И.* Сжатия конфигураций в пространствах знаний // Экологический вестник научных центров Черноморского экономического сотрудничества. 2008, №4. С. 26–33.
2. *Baader F.* Logic-Based Knowledge Representation //Artificial intelligence today. 1999, LNAI 1600. P. 13-41.
3. *Gupta A., Nashimura N.* Finding largest subtrees and smallest supertrees // Algorithmica. 1998. Vol. 21. P. 183–210.
4. *Ахо А., Хопрокрофт Д., Ульман Д.* Структуры данных и алгоритмы. М.: Вильямс, 2003. 382 с.

Ключевые слова: семантическая структура, трассирование структур, сложность алгоритма

Статья поступила 10 марта 2010 г.

Кубанский государственный университет, г. Краснодар