

сегменты Ethernet не рекомендуется нагружать свыше 30% (см. рис. 4). Даже если среднее значение коэффициента использования находится в норме, но имеются пиковые значения, превышающие 60%, то это является свидетельством того, что сеть работает ненормально и требует проведения дополнительных исследований.

Заключение

1. По результатам проведенных исследований эффективной пропускной способности каналов связи теоретически (в пакете Opnet Modeler) и экспериментально (соединением двух компьютеров при помощи коммутатора, а также витой пары типа «cross») можно сделать вывод о том, что результаты исследования имеют расхождение. Эти расхождения совсем незначительные (относительная погрешность менее 4%) и могут

быть использованы для сравнения результатов аналогичных исследований пропускной способности каналов связи.

2. Выбранные протоколы канального уровня (Ethernet, Fast Ethernet, Gigabit Ethernet) были выбраны для исследований в связи с тем, что принципы их работы реализованы в пакете OPNET Modeler, а также они являются наиболее доступными для практического использования.

Литература

1. Тарасов В.Н., Бахарева Н.Ф., Коннов А.Л., Ушаков Ю.А. Проектирование и моделирование сетей ЭВМ в системе Opnet Modeler. Самара: Изд. ПГАТИ, 2008. – 233 с.
2. Уилсон Э. Мониторинг и анализ сетей: Методы выявления неисправностей: Пер. с англ.-М.: ЛОРИ, 2002. – 350 с.

STUDY OF THE EFFECTIVE BANDWIDTH CHANNELS OF COMMUNICATION

Tarasov V.N., Shatov L.G.

The results of studies of effective bandwidth of communication channels, obtained by simulation and real experiments for the equipment.

Keywords: nominal and effective capacity, technology, Ethernet, Fast Ethernet and Gigabit Ethernet.

Тарасов Вениамин Николаевич, д.т.н. профессор, заведующий Кафедрой программного обеспечения и управления в технических системах (ПОУТС) Поволжского государственного университета телекоммуникаций и информатики (ПГУТИ). Тел. (8-846) 228-00-13. E-mail: vt@ist.psati.ru

Шатов Леонид Георгиевич, аспирант Кафедры ПОУТС ПГУТИ. Тел. (8-846) 228-00-13. E-mail: leonid2009@hotmail.com

ТЕХНОЛОГИИ КОМПЬЮТЕРНЫХ СИСТЕМ И СЕТЕЙ

УДК 519.71

ОПТИМАЛЬНАЯ ИНФОРМАЦИОННО-ГРАФОВАЯ МОДЕЛЬ ОРГАНИЗАЦИИ ВКЛЮЧАЮЩЕГО ПОИСКА В ДЕСКРИПТОРНЫХ ПОИСКОВЫХ СИСТЕМАХ

Лялин В.Е., Мальцев С.А., Тарануха В.П., Шишов Д.Р.

В статье рассматривается создание оптимальной информационно-графовой (ИГ) модели организации включающего поиска в дескрипторных поисковых системах для организации непрерывного резервного копирования информации. Проведена оценка сложности включающего поиска, рассмотрены оптимальные информационные графы с недревовидной и древовидной структурой. Приведен подробный алгоритм решения задачи включающего поиска путем построения информационного графа.

Ключевые слова: информационный граф, включающий поиск, множества.

Введение

В общем случае включающий поиск встречается всегда, когда элементы информационного массива задаются множеством свойств (в частности, свойствами наличия дескриптора в описании документа), причем необходимо перечислить в этом массиве элементы с определенным набором свойств задаваемым запросом. В нашем случае описанием объекта файловой системы может служить: название, время резервирования, тип файла и т.д. Особенностью алгоритма включаю-

щего поиска является то, что его использование не подразумевает обязательное упорядочивание базы данных системы резервного копирования по какому-либо признаку. Это полезное свойство можно использовать для организации непрерывности резервирования и восстановления в случае сбойных ситуаций без дополнительных временных затрат и привлечения дополнительных ресурсов по памяти при упорядочивании файловой истории после записи каждого объекта.

Задача включающего поиска

Предположим, что мы осуществляем поиск в дескрипторных поисковых системах, то есть информационный массив состоит из документов, описанных множеством ключевых слов (дескрипторов) [1]. В свою очередь, запрос задает некоторую совокупность дескрипторов, и необходимо перечислить все документы, содержащие в своем описании все дескрипторы, входящие в запрос. Такой тип называется типом включающего поиска.

Занумеруем некоторым образом множество всех дескрипторов (тезаурис). Каждому документу сопоставим запись, представляющую собой булев вектор длины равной мощности тезауруса, в i -ой компоненте которого стоит «0» (ноль) в том и только в том случае, когда i -ый дескриптор входит в описание данного документа. Запросы будем описывать аналогичными векторами, то есть в i -ой компоненте запроса будет стоять «0», если i -ый дескриптор входит в запрос.

Теперь дадим определение типа включающего поиска. Для этого рассмотрим следующий тип задач поиска:

$$S_{bool} = \langle B^n, B^n, \succeq, \sigma, P \rangle,$$

где B^n – единичный n -мерный куб, то есть

$$B^n = \{(\alpha_1, \dots, \alpha_n) : \alpha_i \in \{0, 1\}; (i = \overline{1, n})\},$$

\succeq – отношение поиска на $B^n \times B^n$, определяемое следующим соотношением:

$$(x_1, \dots, x_n) \succeq (y_1, \dots, y_n) \Leftrightarrow x_i \geq y_i, i = \overline{1, n}, x_i, y_i \in \{0, 1\},$$

где i – номер дескриптора. Таким образом, каждый компонент запроса не меньше соответствующей компоненты записи, или если в i -ой компоненте запроса стоит 0 (то есть i -ый дескриптор входит в запрос), то в i -ой компоненте записи тоже должен стоять 0 (то есть запись должна содержать i -ый дескриптор); σ – алгебра подмножеств B^n , пред-

ставляющая собой множество всех подмножеств B^n ; P – равномерная вероятностная мера на σ , то есть такая мера, что для любого $x \in B^n$ выполнено $P(x) = 1/2^n$ и для любого $A \subseteq B^n$ выполнено $P(A) = |A|/2^n$.

Задачи информационного поиска (ЗИП), принадлежащие данному типу, есть разновидность задач, именуемых задачами включающего поиска, поэтому тип S_{bool} называется типом включающего поиска, а задачи, принадлежащие этому типу, – задачами включающего поиска.

Рассмотрим функцию $K_a : X \rightarrow \{0, 1\}$, такую, что $N_{K_a} = \{x \in X : x \succeq a\}$, то есть характеристическую функцию (ХФ) записи a . Множество всех конъюнкций ХФ всех записей из X обозначим через M . Таким образом

$$M = \left\{ \bigvee_{i=1}^n K_{a_i}(x) : a_i \in X \right\}.$$

Подмножество $\{\beta_1, \dots, \beta_m\}$ вершин предикатного информационного графа (ПИГ) U назовем характерным, если существует такая запись a , что $\bigvee_{i=1}^m \phi_{\beta_i} = K_a$. Если в ПИГ U существует цепь, ведущая из корня в какую-либо вершину характерного подмножества, такая, что проводимость этой цепи равна K_a , то эта цепь называется главной для данного подмножества.

Пусть U – произвольный ПИГ над базовым множеством $\tilde{F} = \langle M, \emptyset \rangle$. Пусть B – произвольное характерное подмножество ПИГ U . Тогда в ПИГ U существует главная цепь множества B [2].

Заметим, что $\bigvee_{\alpha \in L_U(y)} \phi_\alpha = \chi_{y \succeq} = K_y$, где $L_U(y)$ – множество всех листьев, которым соответствует запись y , а значит, $L_U(y)$ есть характерное множество. Таким образом, справедливо следующее утверждение.

Пусть $I = \langle X, V, \succeq \rangle$ – задача типа S_{bool} . Пусть U – ПИГ над базовым множеством $\tilde{F} = \langle M, \emptyset \rangle$, разрешающий ЗИП I . Тогда для любой записи $y \in V$ в графе U существует главная цепь.

Назовем гранью единичного n -мерного куба B^n множество

$$\{(\alpha_1, \dots, \alpha_n) \in B^n : \alpha_i = \sigma_1, \dots, \alpha_k = \sigma_k\},$$

при этом число $n - k$ называется размерностью этой грани. Весом набора $\alpha = (\alpha_1, \dots, \alpha_n)$ называют число его координат равных 1. Множество вершин куба, имеющих вес k , называется k -ым слоем куба B^n и обозначается B_k^n . Номером набора α назовем число

$$\|\alpha\| = \sum_{i=1}^n 2^{n-i} \alpha_n.$$

Будем считать, что наборы в слое упорядочены по убыванию номеров. Множество, состоящее из m первых наборов k -го слоя, будем называть начальным отрезком длины m этого слоя.

Формула $x_i^{\sigma_1} \& x_i^{\sigma_2} \& \dots \& x_i^{\sigma_r}$, где $\&$ – знак конъюнкции, $\sigma_k \in \{0, 1\}$, $x_i^0 = \overline{x_{ik}}$, $x_i^1 = x_{ik}$, $i_k \in \{1, 2, \dots, n\}$, $k = 1, 2, \dots, r$, ($r \geq 1, n \geq 1$), называется конъюнкцией над множеством переменных $X^n = \{x_1, x_2, \dots, x_n\}$, а число r – длиной этой конъюнкции. Если $x_{ij} \neq x_{ik}$ при $j \neq k$, то конъюнкция называется элементарной. Элементарная конъюнкция называется монотонной, если она не содержит отрицаний переменных. Множество элементарных монотонных конъюнкций от n переменных будем обозначать через K^n . Функцию тождественная единица будем считать элементарной монотонной конъюнкцией длины 0, то есть будем считать, что она входит в множество K^n , кроме того, добавим тождественную единицу и в множество переменных X^n .

Две элементарные монотонные конъюнкции пересекаются по переменным, если в формулах, описывающих эти конъюнкции, встречаются одинаковые переменные. Функция алгебры логики $f(x_1, \dots, x_n)$ называется монотонной, если для любых двух наборов α и β из B^n таких, что $\alpha \succeq \beta$, имеет место неравенство $f(\alpha) \geq f(\beta)$. Дизъюнкция элементарных монотонных конъюнкций есть монотонная функция. Множество монотонных булевых функций от n переменных будем обозначать через M^n .

Рассмотрим произвольную запись $y \in B^n$. Нетрудно заметить, что если $\{i_1 \dots i_k\}$ есть множество номеров координат вектора y равных 1, то характеристическая функция записи является элементарной монотонной конъюнкцией

$$\chi_{y, \succeq}^b(x_1, \dots, x_n) = x_{i_1} \& \dots \& x_{i_k}.$$

Таким образом, согласно [3], каждое из базовых множеств $\langle M^n, \emptyset \rangle$, $\langle K^n, \emptyset \rangle$ и $\langle X^n, \emptyset \rangle$ является полным для типа S_{bool} , а значит, множество $\tilde{U}(I, \tilde{F})$ ИГ, разрешающих задачу включающего поиска, не пусто. Следовательно, тип S_{bool} полностью удовлетворяет необходимым условиям, и, значит, для любой ЗИП I типа S_{bool} существует оптимальный ПИГ над любым из базовых множеств $\langle M^n, \emptyset \rangle$, $\langle K^n, \emptyset \rangle$ и $\langle X^n, \emptyset \rangle$.

Оценка сложности включающего поиска

Оценим сложность включающего поиска. Для этого рассмотрим теорему Краскала-Като-

ны, переформулировав ее в терминах ИГ модели [4].

Теорема Краскала-Катоны. Пусть B_m^n – m -ый слой куба B^n и $A \subseteq B_m^n$. Пусть $R(A) = \{\alpha \in B_{m+1}^n : (\exists \beta \in A : \alpha \succeq \beta)\}$. Тогда минимум $|R(A)|$ достигается, если A – начальный отрезок слоя B_m^n , причем в этом случае $R(A)$ – начальный отрезок слоя B_{m+1}^n . Если $A \in B^n$, то обозначим $O(A, \succeq) = \bigcup_{y \in A} O(y, \succeq)$.

Приведем следствие из теоремы Краскала-Катоны. Пусть $A \subseteq B_m^n$, тогда $|O(A, \succeq)|$ достигает минимума, если A – начальный отрезок слоя B_m^n .

Пусть $I = \langle B^n, V, \succeq \rangle$ – ЗИП типа S_{bool} . Пусть U – разрешающий ЗИП I ПИГ над базовым множеством $\tilde{F} = \langle M^n, \emptyset \rangle$. Тогда существует такое разбиение библиотеки V на непересекающиеся части, что

- каждая часть содержит записи одного веса, и этот вес назовем весом части;
- каждой части веса $m > 0$ и мощности t можно поставить в соответствие такую совокупность ребер графа U , что сумма сложностей ребер из этой совокупности не меньше чем $t \cdot 2^{1-m}$, причем образы различных частей при этом соответствии не пересекаются.

Доказательство приводится в [5]. Следовательно, можно получить следующие оценки сложности включающего поиска.

Нижняя оценка. Пусть $I = \langle B^n, V, \succeq \rangle$ – ЗИП типа S_{bool} . Пусть базовое множество имеет вид $\tilde{F} = \langle M^n, \emptyset \rangle$, где M^n – множество монотонных булевых функций. Тогда

$$T(I, F) \geq 2 \sum_{y \in V} P(O(y, \succeq)) - t_0,$$

где t_0 – число записей веса 0 в библиотеке V ($t_0 \in \{0, 1\}$). Таким образом, полученная нижняя оценка практически в два раза лучше мощностной нижней оценки.

Верхняя оценка. Пусть $I = \langle B^n, V, \succeq \rangle$ – ЗИП типа S_{bool} . Пусть базовое множество имеет вид $F = \langle K^n, \emptyset \rangle$, где K^n – множество монотонных элементарных конъюнкций. Пусть $|V| = k$ и m – такое число, что $2C_n^m < k \leq 2C_n^{m+1}$. Тогда

$$T(I, \tilde{F}) \leq 1 + \sum_{i=1}^m 2^{1-i} C_n^i + 2^{-m} k.$$

Это говорит о том, что существуют библиотеки, для которых нижняя оценка асимптотически неулучшаема.

Если базовое множество имеет вид $\tilde{F} = \langle F, \emptyset \rangle$, где $F \subseteq M^n$ и $K^n \subseteq F$, то существуют такие $I = \langle B^n, V, \underline{z} \rangle$ типа S_{bool} , что

$$T(I, \tilde{F}) = 2 \sum_{y \in I} P(O(y, \underline{z})) (1 + o(1)).$$

Приведенная нижняя оценка включающего поиска в два раза лучше мощностной нижней оценки. В случае когда базовое множество состоит из множества переменных X^n , удастся найти такие задачи, для которых была получена нижняя оценка, большая по порядку, чем мощностная нижняя оценка. Доказательство этого факта проходит только в предположении древовидности графа, поэтому для начала рассмотрим, являются ли оптимальные ИГ над библиотеками баз данных резервного копирования древовидными в общем случае.

Недревовидность оптимальных ИГ включающего поиска в общем случае

Рассмотрим информационные графы над двумя базовыми множествами: базовым множеством $\langle X^n, \emptyset \rangle$, которое будем называть базовым множеством переменных, и базовым множеством $\langle K^n, \emptyset \rangle$, которое будем называть базовым множеством элементарных монотонных конъюнкций. Для каждого из этих базовых множеств доказано существование таких задач включающего поиска, для которых удалось найти недревовидные ПИГ, которые по сложности проще любого древовидного ПИГ.

Рассмотрим сначала два свойства ПИГ над $\langle X^n, \emptyset \rangle$ или $\langle K^n, \emptyset \rangle$ [6].

1. Пусть U произвольный ПИГ над базовым множеством $\langle X^n, \emptyset \rangle$ или $\langle K^n, \emptyset \rangle$, разрешающий некоторую задачу типа S_{bool} . Тогда любое ребро ПИГ U имеет сложность, не меньшую, чем 2^{-n} .

2. Пусть U произвольный ПИГ над базовым множеством $\langle X^n, \emptyset \rangle$ или $\langle K^n, \emptyset \rangle$, разрешающий некоторую задачу I типа S_{bool} и имеющий вид дерева, в некоторой главной цепи записи которого встречаются два разных ребра, которым приписан один и тот же символ. Тогда граф U не может быть оптимальным для рассматриваемой задачи [5].

Рассмотрим графы над базовым множеством $\langle X^n, \emptyset \rangle$ и покажем существование такой ЗИП типа S_{bool} , которая имеет недревовидный оптимальный ПИГ.

Если $n > 9$ и базовое множество есть множество переменных $\langle X^n, \emptyset \rangle$, то существует такая ЗИП типа S_{bool} , для которой любой оптимальный ПИГ над $\langle X^n, \emptyset \rangle$ не является деревом [7-8]. Очевидно,

что данное утверждение соответствует рассматриваемому нами случаю в системе резервного копирования.

Базовое множество элементарных монотонных конъюнкций. Будем рассматривать графы над базовым множеством $\langle K^n, \emptyset \rangle$ и покажем, что существует такая ЗИП типа S_{bool} , которая имеет недревовидный оптимальный ПИГ. Заметим, что ЗИП I_1 , приведенная в предыдущем подпункте, не дает желаемого результата для базового множества элементарных монотонных конъюнкций. В самом деле, подсчитаем сложность графа U_0 , изображенного на рис. 1, но уже понимая (как и везде далее в этом подпункте) под ребром, которому приписан символ конъюнкции, не цепочку ребер а именно одно ребро с нагрузкой в виде этой конъюнкции.

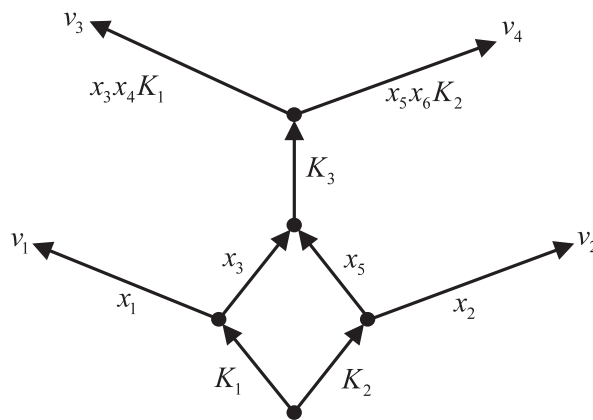


Рис. 1. Граф U_0

Итак,

$$T(U_0) = 2 + 4 \cdot \frac{1}{2^l} + \left(\frac{2}{2^{l+1}} - \frac{1}{2^{2l+2}}\right) + 2 \left(\frac{2}{2^{l+m+1}} - \frac{1}{2^{2l+m+2}}\right) = \frac{1}{2^{2l+m+1}} \times (2^{2l+m+2} + 2^{l+m+3} + 2^{l+m+1} - 2^{m-1} + 2^{l+2} - 1)$$

Сложность графа U_2 , изображенного на рис. 2, будет равна

$$T(U_2) = 2 + \frac{4}{2^l} = \frac{1}{2^{2l+m+1}} \cdot (2^{2l+m+2} + 2^{l+m+3}),$$

откуда

$$T(U_0) - T(U_2) = \frac{1}{2^{2l+m+1}} \cdot (2^{l+m+1} - 2^{m-1} + 2^{l+2} - 1).$$

Очевидно, что $T(U_0) > T(U_2)$ для любых $l, m > 0$.

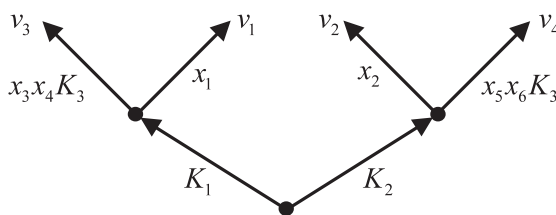


Рис. 2. Граф U_2

Покажем два полезных свойства.

1. Пусть U произвольный ПИГ над базовым множеством $\langle K^n, \emptyset \rangle$, разрешающий некоторую задачу типа S_{bool} , такой, что в нем есть две главные цепи, которые не пересекаются ни с какими другими цепями и которые соответствуют записям, имеющим, по крайней мере, две общие переменные. Тогда граф U не может быть оптимальным.

2. Пусть U произвольный ПИГ над базовым множеством $\langle K^n, \emptyset \rangle$, разрешающий некоторую задачу типа S_{bool} , такой, что в нем есть вершина, отличная от корня, с полустепенью исхода 1. Тогда ПИГ U не может быть оптимальным.

Если $n > 26$ и базовое множество есть множество элементарных монотонных конъюнкций $\langle K^n, \emptyset \rangle$, то существует такая ЗИП типа S_{bool} , для которой любой оптимальный ПИГ над $\langle K^n, \emptyset \rangle$ не является деревом [7].

Древовидность оптимальных ИГ включающего поиска в классе бесповторных сетей

Было показано, что оптимальные графы над базовыми множествами $\langle X^n, \emptyset \rangle$ и $\langle K^n, \emptyset \rangle$, разрешающие ЗИП типа S_{bool} , в общем случае не являются древовидными. Однако существует особый класс ИГ разрешающих ЗИП типа S_{bool} , для которых была доказана их древовидность. Это так называемые бесповторные графы.

Под бесповторными графами понимается ПИГ над базовым множеством $\langle X^n, \emptyset \rangle$ или над базовым множеством $\langle K^n, \emptyset \rangle$, в которых для любой цепи нагрузки любых двух различных ребер этой цепи не пересекаются по переменным [9]. Пусть \tilde{F} одно из двух базовых множеств $\langle K^n, \emptyset \rangle$ или $\langle X^n, \emptyset \rangle$, тогда для любой ЗИП I типа S_{bool} граф над базовым множеством \tilde{F} , являющийся оптимальным в классе бесповторных графов для задачи I , имеет вид дерева [10].

Алгоритмы включающего поиска

Рассмотрим метод решения задачи включающего поиска, имеющий три модификации в зависимости от выбранного базового множества: множества монотонных булевых функций $\langle M^n, \emptyset \rangle$, множества элементарных монотонных конъюнкций $\langle K^n, \emptyset \rangle$ и множества базовых переменных $\langle X^n, \emptyset \rangle$. С помощью этого метода получены оценки функций $T(k, S_{bool}, \langle F, \emptyset \rangle)$ и $\tilde{T}(k, S_{bool}, \langle F, \emptyset \rangle)$, где $F \in \{M^n, K^n, X^n\}$. Кроме того, получена асимптотика логарифма среднего значения сложности данного метода. В данном

разделе предлагаются три алгоритма, которые позволяют для произвольно взятой библиотеки $V \subseteq B^n$ строить информационные графы над базовыми множествами $\langle M^n, \emptyset \rangle$, $\langle K^n, \emptyset \rangle$, $\langle X^n, \emptyset \rangle$, разрешающие ЗИП $I = \langle B^n, V, \succeq \rangle$. Эти алгоритмы будем обозначать соответственно P_1 , P_2 , P_3 .

Поскольку в задаче включающего поиска множество B^n и отношение \succeq фиксированы, то будем считать, что алгоритм P_i ($i = 1; 2; 3$) применяется к библиотеке V и результатом его применения является разрешающий задачу $I = \langle B^n, V, \succeq \rangle$ ИГ, который обозначим через $P_i(V)$. Поскольку ИГ $P_i(V)$ существенно будет зависеть от порядка следования элементов в библиотеке V , то будем считать, что библиотеки – упорядоченные множества. Каждый ИГ $P_i(V)$ ($i = 1; 2; 3$) будет иметь вид дерева, и мы, используя в дальнейшем термин «дерево», будем подразумевать некоторую его укладку, причем будем считать, что в каждой вершине среди ребер, исходящих из данной вершины, задан некоторый порядок следования, и первое по порядку ребро будем называть самым левым, а последнее – самым правым, и соответственно будем определять понятия «левее», «правее».

Будем говорить, что функция f покрывает функцию g , если $N_g \subseteq N_f$. Пусть f, g – элементарные монотонные конъюнкции. Пересечением конъюнкций f и g назовем конъюнкцию переменных, принадлежащих как f , так и g , и обозначим ее через $f \cap g$. Таким образом, пересечение f и g есть минимальная по числу элементов в характеристическом множестве элементарная монотонная конъюнкция, покрывающая и f , и g . В случае, когда все переменные f отличны от переменных g , будем говорить, что пересечение f и g пусто. Через $f \setminus g$ обозначим конъюнкцию переменных, принадлежащих f , но не принадлежащих g .

Обозначим через $\tilde{0}$ набор, состоящий из всех нулей, то есть $\tilde{0} = (0, \dots, 0)$. Пусть нам дана упорядоченная библиотека V . Обозначим $V' = V \setminus \{\tilde{0}\} = \{y_1, y_2, \dots, y_k\}$. Индуктивно опишем процесс построения ИГ $P_2(V')$.

Базис индукции. $P_2(\{y_1\})$ состоит из одного ребра, начало которого является корнем ИГ, а конец – листом, которому приписана запись y_1 , и этому ребру приписана функция $\chi_{y_1, \succeq}$.

Индуктивный переход. Пусть построен ИГ $P_2(\{y_1 \dots y_{i-1}\})$, ($i = 2; 3 \dots k$). Будем перестраивать его, чтобы он стал допустимым для библиотеки $\{y_1 \dots y_i\}$, следующим образом.

I. Корень ИГ $P_2(\{y_1, \dots, y_{i-1}\})$ объявим в качестве текущей вершины β . Объявим $g = \chi_{y_i, \geq}^b$.

II. Будем слева направо просматривать ребра, исходящие из вершины β , до тех пор пока не найдем ребро, пересечение нагрузочной функции которого с функцией g не пусто.

1. Если для всех ребер пересечение их нагрузочных функций с g пусто, то из вершины β выпускаем новое ребро с нагрузочной функцией g , конец ребра объявляем листом и приписываем ему запись y_i , и на этом завершаем работу – ИГ $P_2(\{y_1, \dots, y_i\})$ построен. Здесь, как и везде далее, новое выпускаемое ребро становится самым правым в пучке на данный момент.

2. Если же нашлось ребро с нагрузочной функцией f , такой, что пересечение f и g не пусто, то:

а) если $f = g$, то конец этого ребра объявляем листом и приписываем ему запись y_i , и на этом завершаем работу – ИГ $P_2(\{y_1, \dots, y_i\})$ построен;

б) если $f \neq g$ и $f = f \cap g$, то конец этого ребра объявляем текущей вершиной β , объявляем $g = g \setminus f$ и идем на шаг II;

в) если же $f \neq f \cap g$, то изменим нагрузку рассматриваемого ребра на $f \cap g$. Пусть конец этого ребра обозначен через β' , отцепим ветвь, растущую из β' , выпустим из β' новое ребро и прицепим отцепленную ветвь к концу нового ребра (если вершина β' была листом, то листом вместе с его нагрузкой становится конец нового ребра) и приписываем новому ребру функцию $f \setminus (f \cap g)$, после чего если $f = g$, то объявляем вершину β' листом, приписываем ей запись y_i , и завершаем работу, иначе из вершины β' выпустим еще одно ребро с нагрузкой $g \setminus (f \cap g)$, конец последнего ребра объявляем листом, приписываем ему запись y_i и на этом завершаем работу – ИГ $P_2(\{y_1 \dots y_i\})$ построен.

Теперь если $V \neq V'$, то выпустим из корня $P_2(V')$ еще одно ребро, и пусть оно будет самым правым, исходящим из корня. Припишем данному ребру функцию тождественную «1», а конец ребра объявим листом и припишем ему запись « $\bar{0}$ ». Полученный ИГ и будет $P_2(V)$.

Заметим, что $P_2(V)$ есть ИГ над базовым множеством $\langle K^n, \emptyset \rangle$ и, кроме того, нагрузка ребер в $P_2(V)$ однозначно определяет нагрузку листьев. Поэтому далее мы часто не будем упоминать о нагрузке листьев.

Обозначим $\chi_{V, \geq}^b = \bigvee_{y \in V} \chi_{y, \geq}^b$. Теперь легко описать ИГ $P_1(V)$. Возьмем вершину и объявим ее корнем, выпустим из нее ребро, которому припишем функцию $\chi_{V, \geq}^b$, а из конца ребра выпустим ветвь, идентичную $P_1(V)$. ИГ $P_1(V)$ построен.

Очевидно, что $P_1(V)$ есть ИГ над базовым множеством $\langle M^n, \emptyset \rangle$.

Опишем процесс построения ИГ $P_3(V')$.

Базис индукции. Если в нашем случае $\chi_{y_i, \geq}^b = x_{i_1} \cdot \dots \cdot x_{i_t}$ ($1 \leq i_1 < \dots < i_t \leq n$), то $P_3(\{y_i\})$ представляет собой ориентированную цепочку из t ребер, причем начало цепочки есть корень ИГ, конец цепочки – лист с записью y_i , а j -му ребру цепочки ($j = 1, 2, \dots, t$) приписана функция x_{i_j} .

Индуктивный переход. Пусть построен ИГ $P_3(\{y_1, \dots, y_{i-1}\})$, ($i = 2, 3, \dots, k$). Будем перестраивать его, чтобы он стал допустимым для библиотеки $\{y_1, \dots, y_i\}$, следующим образом.

I. Корень ИГ $P_3(\{y_1, \dots, y_{i-1}\})$ объявим в качестве текущей вершины β . Объявим $g = \chi_{y_i, \geq}^b$.

II. Будем слева направо просматривать ребра, исходящие из вершины β , до тех пор, пока не найдем ребро, которому соответствует переменная, покрывающая g .

1. Если для всех ребер, исходящих из β , переменная, соответствующая ребру, не покрывает g , то из вершины β выпускаем новую цепочку ребер, аналогичную цепочке, описанной в базисе индукции, такую, чтобы конъюнкция переменных, приписанных ребрам цепочки, была равна g , конец цепочки объявляем листом, приписываем ему запись y_i и завершаем работу – ИГ $P_3(\{y_1 \dots y_i\})$ построен. Здесь первое ребро новой цепочки становится самым правым, исходящим из β .

2. Если же нашлось ребро с нагрузочной функцией x_j , покрывающей g , то если $g \neq x_j$, то конец этого ребра объявляем текущей вершиной β , объявляем $g = g \setminus x_j$ и идем на шаг II, иначе конец рассматриваемого ребра объявляем листом, приписываем ему запись y_i и завершаем работу – ИГ $P_3(\{y_1 \dots y_i\})$ построен.

Теперь если $V \neq V'$, то выпустим из корня $P_3(V')$ еще одно ребро, и пусть оно будет самым правым, исходящим из корня. Припишем данному ребру функцию, тождественную 1, а конец ребра объявим листом и припишем ему запись $\bar{0}$. Полученный ИГ и будет $P_3(V)$.

Легко заметить, что по построению каждого из ИГ $P_1(V)$, $P_2(V)$, $P_3(V)$ к произвольной записи $y \in V$ ведет цепочка ребер, проводимость которой равна $\chi_{y, \geq}^b$. Отсюда согласно [11] каждый из ИГ $P_1(V)$, $P_2(V)$, $P_3(V)$ допустим для библиотеки V .

Рассмотрим теперь верхние и нижние оценки сложностей графов $P_1(V)$, $P_2(V)$ и $P_3(V)$. Через W_n^k обозначим множество упорядоченных библиотек из B^n мощности k . Понятно, что

$|W_n^k| = A_{2^n}^k$, где A_n^k – число размещений из n элементов по k . Обозначим через $\tilde{T}(P_i, k, n)$ максимальную из сложностей ЗИП над всеми упорядоченными библиотеками мощности k , то есть $\tilde{T}(P_i, k, n) = \max_{V \in W_n^k} T(P_i(V))$, а через $T(P_i, k, n)$ – среднюю сложность ЗИП над библиотекой мощности k :

$$T(P_i, k, n) = \frac{\sum_{V \in W_n^k} T(P_i(V))}{|W_n^k|}, \quad i = 1; 2; 3.$$

Поскольку все $P_i(V)$ ($i = 1; 2; 3$) являются деревьями, то далее часто вместо термина «ИГ» будем использовать термин «дерево».

Следом вершины β в дереве D назовем поддеревом дерева D , содержащее все ребра, исходящие из вершин, принадлежащих пути, соединяющему корень дерева D с вершиной β . Левым следом вершины β в дереве D назовем поддерево дерева D , которое состоит из принадлежащих следу вершины β ребер, которые в дереве D находятся не правее ребра, принадлежащего пути, соединяющему корень дерева D с вершиной β . Библиотекой вершины β в дереве D назовем множество записей, приписанных листьям, достижимым из вершины β . Вершина дерева называется проходной, если из нее исходит одно ребро.

Рассмотрим некоторые полезные свойства деревьев $P_2(V)$ и $P_3(V)$. Пусть V – произвольная библиотека, $D = P_i(V)$ ($i = 2; 3$), тогда это дерево обладает следующими свойствами.

1. Для любой вершины β дерева D пересечение любых двух нагрузочных функций, соответствующих ребрам, принадлежащих левому следу вершины β , всегда пусто.

2. Функции фильтров любых двух различных вершин в D различны.

3. В дереве $P_2(V)$ каждая проходная вершина является полюсом.

Эти свойства достаточно очевидно следуют из алгоритмов построения $P_2(V)$ и $P_3(V)$.

Напомним, что сложностью вершины β в ИГ называется вероятность множества запросов, проходящих в вершину β , или, иными словами, сложность вершины β равна $P(N_{\beta_p})$. Высотой вершины в дереве назовем длину пути, ведущего из корня в данную вершину.

Пусть D – некоторое дерево, h – натуральное число. Через $r(D, h)$ обозначим число вершин дерева D , находящихся на высоте h . Обозначим

$$D_i = \{P_i(V) : V \in W_n^k\}, \quad i = 1; 2; 3.$$

Вершину ИГ назовем внутренней, если она не является полюсом.

Пользуясь определениями деревьев $P_i(V)$ ($i = 1; 2; 3$), несложно показать справедливость следующих утверждений:

Для любой вершины, находящейся на высоте h в любом дереве $D \in D_i$ ($i = 2; 3$), ее сложность не более чем 2^{-h} .

Для любого дерева

$$D \in D_i, \quad i = 2; 3, \quad r(D, 1) \leq C_n^1 + 1,$$

и для любого $h \in \{2 \dots n\}$ справедливо $r(D, h) \leq C_n^h$.

Пусть $D \in D_2$, t_j – число листьев в дереве D , находящихся на высоте j ($j \in \{1, \dots, n\}$). Тогда для любого $h \in \{0 \dots n-1\}$ справедливо

$$r(D, n-h) \leq \sum_{j=0}^h 2^{j-h} t_{n-j}.$$

Рассмотрим верхнюю и нижнюю оценки для алгоритмов $P_1(V)$ и $P_2(V)$. Обозначим $\tilde{F}_1 = \langle M^n, \emptyset \rangle$, $\tilde{F}_2 = \langle K^n, \emptyset \rangle$. Если $k(n)$ и $m(n)$ такие числа, зависящие от n , что $k(n) \rightarrow \infty$, $m(n) = o(n)$ при $n \rightarrow \infty$, то для $i = 1; 2$:

1. При $C_n^{m-1} = o(k)$, $k \leq C_n^m$:

$$\tilde{T}(k, S_{bool}, \tilde{F}_i) \sim \tilde{T}(P_i, k, n) \sim 2^{1-m} k.$$

2. При $C_n^m \leq k \leq 2C_n^m$:

$$2^{-m}(k + C_n^m) \leq \tilde{T}(k, S_{bool}, \tilde{F}_i) \leq \tilde{T}(P_i, k, n) \leq 2^{1-m} k(1 + o(1)).$$

3. При $k = O(C_n^m)$, $k \geq 2C_n^m$:

$$2^{-m}(k + C_n^m) \leq \tilde{T}(k, S_{bool}, \tilde{F}_i) \leq \tilde{T}(P_i, k, n) \leq 2^{1-m}(k + C_n^m)(1 + o(1)).$$

Таким образом, для задач включающего поиска удалось получить нижнюю оценку в два раза лучшую, чем мощностная нижняя оценка. Кроме того, было доказано существование задач, для которых эта нижняя оценка асимптотически не улучшаема.

Литература

1. Черный А.И. Введение в теорию информационного поиска. М.: Наука, 1975. – 238 с.
2. Bayer R., Mc-Creight E. M. Organization and Maintenance of Large Ordered Indexes // Ada Informatica. Vol. 1, No 3, 1972. – P. 173-189.
3. Карцев М.А. Распараллеливание алгоритмов итерационного типа // Вопросы радио-

- электроники. Сер. ЭВТ. Вып. 9, 1971. – С. 36-39.
4. Вальковский В.А., Котов В.Е., Милошко Й. Алгоритмы, математическое обеспечение и архитектура многопроцессорных вычислительных систем. М.: Наука, 1982. – 336 с.
 5. Гасанов Э.Э., Кудрявцев В.Б. Теория хранения и поиска информации. Москва: Физмат-лит, 2002. – 288 с.
 6. Решетников В. Н. Моделирование информационного поиска в информационно-поисковых системах // Кибернетика. № 5, 1079. – С. 129-132.
 7. Ким Д.П. Методы поиска и преследования подвижных объектов. М.: Наука, 1989. – 336 с.
 8. Кнут Д.Э. Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск. М.: Мир, 1978. – 355 с.
 9. Колмогоров А.Н., Успенский В.А. К определению алгоритма / Успехи математических наук. Т.13, № 4, 1958. – С. 3-28.
 10. Гасанов Э.Э. О сложности поиска в базах данных // Искусственный интеллект. Межвузовский сборник трудов. Саратов: Изд. СГУ, 1993. – С. 41-56.
 11. Носков В.Н. О сложности тестов, контролирующей работу тестов логических схем // Математические заметки. Т. 18, № 1, 1975. – С. 137-150.

OPTIMAL INFORMATION-GRAPH MODEL ORGANIZATION INCLUSIVE SEARCH IN DESCRIPTOR SEARCH ENGINES

Ljalin V.E., Maltzev S.A., Taranucha V.P., Shishov D.R.

This article describes how to create an optimal information-graph model of the organization of inclusive search in descriptor search engine to provide continuous backup information. The evaluation of inclusive search considered optimal information and graphs with nedrevovidnoy tree structure. A detailed algorithm for solving the problem of inclusive search by building information graph.

Keywords: information graph, IG, including search and sets.

Лялин Вадим Евгеньевич, д.т.н., д.э.н., профессор, Заслуженный изобретатель РФ, заведующий Кафедрой математических технологий в нефтегазовом машиностроении (МТНМ) Ижевского государственного технического университета им. М.Т. Калашникова (ИжГТУ). Тел. 8-912-767-30-00. Email: velyalin@mail.ru

Мальцев Сергей Андреевич, к.т.н., ведущий инженер ЗАО «Группа Компаний Старт» (г. Москва). Тел. 8-929-682-90-19. E-mail: rshavas@mail.ru

Тарануха Владимир Прокофьевич, к.т.н., доцент, заведующий Кафедрой конструирования радиоэлектронной аппаратуры ИжГТУ. Тел. 8-341-250-40-22. Email: kra_dept@istu.ru

Шишов Дмитрий Родионович, аспирант Кафедры МТНМ ИжГТУ. Тел. 8-912-447-51-40. E-mail: d1men27@mail.ru

УДК 681.3.06(07)

ПРОТОТИП УНИВЕРСАЛЬНОГО ПРОГРАММНОГО ИНСТРУМЕНТА ДЛЯ ПОСТРОЕНИЯ МАШИННОГО КОММЕНТАРИЯ В ВИДЕ ДИАГРАММЫ КЛАССОВ ИСХОДНЫХ ТЕКСТОВ ПРОГРАММ

Зубов М.В., Пустыгин А.Н., Старцев Е.В.

В статье описывается использование промежуточных представлений исходного кода для выполнения статического анализа. Предлагается расширить использование универсальных и многоуровневых представлений в одном общем. Представлен прототип программного инструмента, генерирующий и обрабатывающий такое представление.

Ключевые слова: статический анализ, промежуточное представление, машинные комментарии, исходный код, диаграмма классов, Java, Python.

Введение

Статический анализ облегчает выполнение типовых задач разработки. Эти задачи могут быть связаны с поиском ошибок, улучшением качества кода, внутренним тестированием [1]. Перспективным направлением является извлечение информации по исходному коду (построение машинных комментариев [2]).