

Программные системы и вычислительные методы

Правильная ссылка на статью:

Пекунов В.В. Объектно-транзакционные модели программ на алгоритмических языках // Программные системы и вычислительные методы. 2024. № 4. DOI: 10.7256/2454-0714.2024.4.69228 EDN: JWFWBE URL: [https://nbpublish.com/library\\_read\\_article.php?id=69228](https://nbpublish.com/library_read_article.php?id=69228)

## Объектно-транзакционные модели программ на алгоритмических языках

Пекунов Владимир Викторович

доктор технических наук

Инженер-программист, ОАО "Информатика"

153000, Россия, Ивановская область, г. Иваново, ул. Ташкентская, 90

 [pekunov@mail.ru](mailto:pekunov@mail.ru)



[Статья из рубрики "Системный анализ, поиск, анализ и фильтрация информации"](#)

### DOI:

10.7256/2454-0714.2024.4.69228

### EDN:

JWFWBE

### Дата направления статьи в редакцию:

05-12-2023

**Аннотация:** Данная работа посвящена вопросу представимости программ, написанных на алгоритмических языках, с помощью формализмов, базирующихся на идее применения предельной частично транзакционной памяти, включающей единственную транзакционную ячейку и множество обычных ячеек. При этом предполагается, что такие формализмы основаны на понятии сети объектов, представляющих как основные, так и вспомогательные элементы решающей задачи. Объекты функционируют в памяти указанного вида, выполняя методы, содержащие исключительно ветвящийся код, лишенный циклов. Циклы при таком подходе заменяются многократным специальным пересогласованием сети объектов, схожим с реализованным в классической транзакционной памяти. Исходя из самых общих представлений о процессе решения задачи в некоторой предметной области, впервые вводится концепция объектно-транзакционной модели и формулируются их основные свойства. При формулировании структуры и основных принципов функционирования объектно-транзакционных моделей используются методы дискретной математики, теории алгоритмов. Вводится понятие предельной частично транзакционной памяти, содержащей единственную

транзакционную ячейку со специальным согласованием. Описываются особенности согласования такой памяти в контексте предлагаемых моделей. Выдвигается гипотеза о реализуемости произвольных алгоритмов с помощью объектно-транзакционных моделей. Описаны основные принципы функционирования таких моделей, сформулированы их основные свойства. Вводятся понятия предельных непараллельной и параллельной моделей. Доказывается, что предельная непараллельная модель способна выполнить произвольный разрешимый по Тьюрингу алгоритм. Доказывается, что предельная параллельная модель из  $K+2$  узлов эквивалентна системе из  $K$  параллельно работающих машин Тьюринга и, соответственно, способна выполнить произвольный разрешимый по Тьюрингу алгоритм, подразумевающий наличие  $K$  параллельных ветвей. Таким образом, выдвинутая в работе гипотеза о реализуемости произвольных алгоритмов доказана.

### **Ключевые слова:**

последовательный алгоритм, модель программы, транзакционная память, специальное согласование, предельные теоремы, объектно-транзакционная модель, машина Тьюринга, теория, параллельный алгоритм, реализуемость алгоритмов

Транзакционная память [1, 2] – достаточно хорошо изученный формализм для построения параллельных программ, отличающийся простотой распараллеливания, связанной с отсутствием необходимости в явных синхронизациях [3]. Теоретически интересным является вопрос, можно ли с помощью некоторого формализма, базирующегося на той или иной предельной модификации транзакционной памяти, строить произвольные (последовательные или параллельные) программы, используя единственную алгоритмическую конструкцию – ветвление. При этом явные циклы можно заменить многократным условным пересогласованием транзакции. Достаточно очевидно, что такая замена требует работы с данными, сохраняющими свои значения при откате транзакции [4], а значит и перехода от полной изоляции транзакций к их частичной изоляции. Такой подход к реализации транзакционной памяти известен [5], в данной работе используется авторский вариант такой памяти [6] – предельная частично транзакционная память, включающая единственную специальную транзакционную ячейку.

Таким образом, целью данной работы является исследование проблемы описания последовательных и параллельных алгоритмов на базе формализмов, основанных на применении предельной частично транзакционной памяти в условиях единственной алгоритмической конструкции ветвления. Такая постановка проблемы является новой. Поставим задачу разработки и исследования свойств такого формализма, исходя из самых общих представлений о концептах, используемых для построения алгоритма, решающего некоторую задачу.

### **Общая постановка проблемы**

Пусть решается некоторая задача в произвольной предметной области. Такая область может быть описана в терминах некоторых понятий, то есть мыслей, обобщающих и выделяющих предметы некоторых классов по общим признакам. Следовательно, предметная область может быть определена как иерархия классов (понятий), описывающих общие свойства объектов, подпадающих под данные понятия. Предположим, что для задачи известен общий, разрешимый по Тьюрингу алгоритм ее решения, причем он может быть декомпозирован на серию многократно пересогласуемых

последовательных и параллельных ациклических процессов (методов объектов, относящихся к классам соответствующей предметной области), последовательность активизации которых определяется по схеме сетевого графика работ.

Тогда формальным описанием процесса решения исходной задачи будет сеть взаимосвязанных объектов, являющихся экземплярами классов соответствующей предметной области, взаимодействующих, например, через общую память. Данное описание может быть как предельным, включающим единственный объект, исчерпывающим образом описывающий процесс решения задачи, так и непредельным, включающим более одного объекта.

Можно предположить, что в самом элементарном случае полное решение задачи выполняется путем однократного запуска процессов сети в требуемом порядке. Однако очевидно, что в общем случае может потребоваться управляемый многократный перезапуск сети, например, в результате пересогласования по некоей специальной транзакционной ячейке. Интересными следствиями наличия такого механизма являются следующие:

- а) если пересогласование по транзакционной памяти не требуется, то сеть всегда завершает работу за конечное время;
- б) с помощью пересогласования в условиях предельной частично транзакционной памяти может быть реализовано неоднократное исполнение одного и того же алгоритма, что является приемлемой альтернативой циклам.

Введем теперь понятие предельной частично транзакционной памяти. Пусть она представляет собой ассоциативный кортеж  $R_K = (V_K, T_K)$ , ключами К которого являются некие идентификаторы, а значениями  $V_K$  могут быть произвольные структуры данных, причем каждая ячейка может иметь собственный тип  $T_K$  со значениями из множества  $\{a, t\}$ . Ячейки типа  $a$  являются элементарными ячейками с атомарным доступом, единственная ячейка типа  $t$  относится к транзакционной памяти, локальные значения которой, фактически, хранятся в векторе  $V_K = (L_1, \dots, L_N)$ . Каждый  $i$ -й элемент такого вектора является двойкой  $(S_i, C_i)$ , где  $S_i$  – метка времени первой модификации ячейки в  $i$ -ом потоке,  $C_i$  – локальная копия ячейки для  $i$ -го потока (всего  $N$  параллельных потоков – параллельных путей исполнения).

Пусть  $C$  – множество классов. Класс  $S$  представляет собой тройку  $(N_c, F, M)$ , где  $N_c$  – идентификатор (имя) класса,  $F$  – поля,  $M$  – методы. Определим отношение иерархии  $parent: C \rightarrow C$ , заданное функцией

$$\forall a \in C \exists parent(a) = \begin{cases} (\epsilon, \emptyset, \emptyset), & \text{если } a \text{ есть корень иерархии;} \\ c \in C, & \text{если } c \text{ есть предок } a, \end{cases}$$

где  $\epsilon$  – пустая цепочка. Пусть каждый метод множества  $M$  представляет собой ветвящийся алгоритм, не содержащий циклов, имеющий полный доступ ко всем полям узла-объекта и ко всем ячейкам некоторой общей предельной частично транзакционной памяти, если таковая присутствует.

**Гипотеза.** Модель, включающая сетевой график  $(P, E)$ , содержащий множество  $P$ , включающее объекты классов  $C$  текущей предметной области (соединенные дугами из множества  $E$ ), вызываемые в определенном выше порядке [с учетом того, что методы объектов, между которыми не определено отношение следования (по структуре модели),

всегда выполняются в параллель], функционирующие по ветвящимся алгоритмам в условиях предельной частично транзакционной памяти со специальным согласованием, способна выполнить произвольный разрешимый по Тьюрингу последовательный или параллельный алгоритм.

Назовем такие модели объектно-транзакционными (ОТМ).

### **Непараллельные ОТМ**

Введем понятие *предельной непараллельной объектно-транзакционной модели*. Пусть такая модель является двойкой  $(R, N)$ , где  $R$  – ассоциативный кортеж предельной частично транзакционной памяти,  $N$  – сеть из единственного узла-объекта  $A$ , относящегося к некоторому классу.

Введем специальный механизм согласования единственной транзакционной ячейки. Пусть она имеет идентификатор `RESTART` и хранит ссылку на текущий исполняемый метод объекта  $A$  (сразу заметим, что в случае ОТМ из нескольких узлов это будет список пар вида  $(A, M[A])$ , где  $A$  – ссылка на узел,  $M[A]$  -- ссылка на метод узла  $A$ ), отличающаяся тем, что

- а) в начале исполнения модели данная ячейка хранит ссылку на специальный метод, всегда вызываемый первым;
- б) любое присваивание данной ячейке ссылки на произвольный метод неминуемо приводит к необходимости пост-пересогласования текущей транзакции (всей сети, по окончании ее текущей интерпретации) с вызовом указанного метода;
- в) перед пост-пересогласованием текущей транзакции определяются подлежащие вызову методы объектов – это методы, ссылки на которые занесены в `RESTART`. Далее ячейка `RESTART` очищается и начинается пост-пересогласование транзакции (всей сети с самого начала).

Отметим, что все ячейки памяти типа  $a$  не влияют на процесс согласования.

**Теорема 1.** Предельная непараллельная ОТМ способна реализовать произвольный разрешимый по Тьюрингу алгоритм.

**Доказательство.** Рассмотрим произвольную программу на алгоритмическом языке высокого уровня С (с расширением, реализующим требуемый механизм согласования предельной частично транзакционной памяти). Если такая программа не содержит циклов, то она может быть реализована произвольным методом (реализующим произвольный ветвящийся алгоритм) единственного объекта предельной непараллельной ОТМ по определению.

Пусть программа содержит циклы. Эквивалентно трансформируем ее по следующим правилам:

- а) свяжем с каждым оператором некую условную метку;
- б) введем нетранзакционную переменную `CURLABEL`, хранящую имя метки текущего оператора;
- в) в начале программы `CURLABEL` содержит метку начального оператора;
- г) каждый оператор с меткой предваряем конструкцией вида

```
[else] if (CURLABEL == метка_текущего_оператора) {
```

д) в каждом операторе с меткой производим соответствующие присваивания переменной CURLABEL меток операторов, выполняемых следующими;

е) в конце каждого оператора ставим закрывающий блок знак «}», перед которым (за исключением операторов, исполняемых последними), ставим конструкцию рассогласования текущей транзакции (приводящую к ее последующему повтору по завершению текущего сеанса обработки сети) путем присваивания переменной RESTART ссылки на текущий метод;

ж) сначала преобразуем все циклы в цикл while;

з) каждый цикл while вида;

```
Метка1: while(условие) { Оператор_A; ... Оператор_B; } Оператор_C;
```

заменим на условные операторы:

```
if (CURLABEL == Метка1 && условие) { CURLABEL = Метка_A; RESTART = ...; }

else if (CURLABEL == Метка1 && !условие) { CURLABEL = Метка_C; RESTART = ...; }

else if (CURLABEL == Метка_A) { Оператор_A; CURLABEL = ...; RESTART = ...; }

...

else if (CURLABEL == Метка_B) { Оператор_B; CURLABEL = Метка_1; RESTART = ... }

else if (CURLABEL == Метка_C) { Оператор_C; ... }
```

и) вся программа заключается в транзакционный блок;

к) все переменные программы оформляем как ячейки типа а предельной частично транзакционной памяти.

Получили эквивалентную программу, которая может быть исполнена произвольным методом ОТМ по определению. Итак, предельная непараллельная ОТМ способна реализовать произвольную программу (например, на алгоритмическом языке высокого уровня С), которой, как известно, может быть программа, реализующая алгоритм функционирования машины Тьюринга (способной реализовать произвольный разрешимый по Тьюрингу алгоритм), следовательно, такая ОТМ способна реализовать произвольный разрешимый по Тьюрингу алгоритм. Также отметим несомненную прямую структурную аналогию между полученной после преобразования программой на С и кодом для машины Тьюринга.

Доказано.

**Следствие.** Поскольку предельная непараллельная ОТМ способна реализовать произвольный, разрешимый по Тьюрингу алгоритм, следовательно, и непредельная непараллельная ОСМ способна реализовать такой алгоритм (например, используя один свой произвольный узел).

### Параллельные ОТМ

Введем понятие предельной параллельной ( $K$ -поточной) объектно-транзакционной модели. Пусть такая модель является двойкой  $(R, N)$ , где  $R$  – ассоциативный кортеж

предельной частично транзакционной памяти,  $N$  – сеть, включающая узлы с пустыми методами (исток и сток), а также  $K$  рабочих узлов с непустыми методами, в каждый из которых входит дуга из истока и из каждого исходит дуга в сток. Пусть модель выполняет транзакционный блок по всем описанным выше правилам (возможно, с многократным пост-пересогласованием), учитывая, что процессы узлов, находящихся в ветвях, для которых не определено отношение следования, выполняются параллельно.

**Теорема 2.** Предельная параллельная ( $K$ -поточная) ОТМ способна реализовать произвольный разрешимый по Тьюрингу алгоритм с  $K$  параллельными ветвями.

**Доказательство.** Выше было показано, что одноузловая непараллельная ОТМ способна реализовать произвольный, разрешимый по Тьюрингу алгоритм. Здесь мы имеем систему из  $K$  узлов (которые могут работать параллельно), каждый из которых можно рассматривать как отдельную ОТМ, взаимодействующую с прочими ОТМ через общую память. Таким образом, всю систему можно рассматривать как  $K$  параллельно работающих машин Тьюринга, обменивающихся данными, например, через общую ленту. Соответственно, данная система способна выполнить произвольный разрешимый по Тьюрингу алгоритм с  $K$  параллельными ветвями.

**Следствие.** Поскольку предельная параллельная ( $K$ -поточная) ОТМ способна реализовать произвольный параллельный  $K$ -поточный алгоритм, разрешимый по Тьюрингу, следовательно и непредельная параллельная ОСМ с не менее чем  $K$  параллельными ветвями способна реализовать такой алгоритм.

### Основные свойства ОТМ

В заключение сформулируем некоторые основные свойства объектно-транзакционных моделей, предложенных в данной работе. Данные свойства вытекают либо из основных структурно-функциональных особенностей моделей, либо из доказанных выше положений. Итак, основные свойства:

1. Предельная единичность. Каждой уникальной программе на некотором алгоритмическом языке высокого уровня соответствует не менее одной эквивалентной ей ОТМ. Данное свойство вытекает из потенциальной реализуемости с помощью ОТМ произвольного разрешимого по Тьюрингу алгоритма (теорема 1). Следовательно, эквивалентом может быть как минимум предельная непараллельная ОТМ, реализующая машину Тьюринга на заданном языке высокого уровня, выполняющую указанный алгоритм. Кроме того, не исключено построение иной реализующей ОТМ, например, подобной по структуре оствовному дереву графа блок-схемы данного алгоритма и содержащей один и более узлов с методами на заданном языке высокого уровня.
2. Алгоритмическая полнота по Тьюрингу (следует из теоремы 1).
3. Естественное описание параллелизма (параллельные и непараллельные процессы четко определяются структурой модели).
4. Атранзакционная конечность. Если механизм согласования единственной ячейки транзакционной памяти не действует, то ОТМ всегда завершает работу за конечное время, поскольку имеет сетевую структуру, что в данном случае делает зацикливание принципиально невозможным.
5. Принципиальная неоднородность используемых видов памяти при неизбыточной (исключающей простое дублирование тел циклов) реализации алгоритмов, содержащих циклы со счетчиком или межвитковыми зависимостями. Следует из того факта, что

модель имеет сетевую структуру, ее методы также имеют ветвящуюся структуру, поэтому такие циклы неизбыточно могут быть реализованы только за счет пересогласования модели (требуется ячейка RESTART предельной частично транзакционной памяти) с сохранением изменяющегося контекста [счетчиков или иных общих для витков цикла переменных (требуются ячейки типа а частично транзакционной памяти)].

## **Выводы**

В данной работе рассмотрена проблема представимости параллельных и непараллельных алгоритмов с помощью объектно-транзакционных моделей, базирующихся на идее предельной частично транзакционной памяти, содержащей единственную транзакционную ячейку со специальным соглашением, в условиях наличия единственной алгоритмической конструкции – ветвления. Впервые сформулировано понятие таких моделей, описаны основные принципы их функционирования, сформулированы их основные свойства. Доказаны теоремы о разрешимости такими моделями произвольных последовательных алгоритмов, вычислимых по Тьюрингу, а также о разрешимости произвольных алгоритмов с К параллельными ветвями, вычислимых системой из К параллельно работающих машин Тьюринга, взаимодействующих через общую ленту.

## **Библиография**

1. Черняк, Л. Транзакционная память-первые шаги / Л. Черняк // Открытые системы. СУБД. – 2007. – № 4. – С. 12-15.
2. Marathe, V.J., Scherer, W.N., Scott, M.L. (2005). Adaptive Software Transactional Memory. In: Fraigniaud, P. (eds) Distributed Computing. DISC 2005. Lecture Notes in Computer Science, vol 3724. Springer, Berlin, Heidelberg.  
[https://doi.org/10.1007/11561927\\_26](https://doi.org/10.1007/11561927_26)
3. M. Herlihy and J. E. B. Moss. Transactional memory: Architectural support for lock-free data structures. In A. J. Smith, editor, Proceedings of the 20th Annual International Symposium on Computer Architecture. San Diego, CA, May 1993, pages 289–300. ACM, 1993.
4. V. Luchangco and V. J. Marathe. Transaction communicators: Enabling cooperation among concurrent transactions. In Proceedings of the 16th ACM Symposium on Principles and Practice of Parallel Programming, PPoPP ’11, pages 169-178, New York, NY, USA, 2011. ACM.
5. Miculan, M., Peressotti, M. Software Transactional Memory with Interactions. Proceedings of the 21st Italian Conference on Theoretical Computer Science, Ischia, Italy, September 14-16, 2020, pp. 67-80.
6. Пекунов В.В. Сверхоптимистичные вычисления: концепция и апробация в задаче о моделировании электростатической линзы // Программные системы и вычислительные методы. – 2020. № 2. – С. 37-44. DOI: 10.7256/2454-0714.2020.2.32232. URL: [https://e-notabene.ru/ppsrm/article\\_32232.htm](https://e-notabene.ru/ppsrm/article_32232.htm)

## **Результаты процедуры рецензирования статьи**

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная на рецензирование научная статья на тему: «Объектно-транзакционные модели программ на алгоритмических языках» представляет собой актуальное исследование.

В рецензируемой статье авторами определена цель исследования - исследование

проблемы описания последовательных и параллельных алгоритмов на базе формализмов, основанных на применении предельной частично транзакционной памяти в условиях единственной алгоритмической конструкции ветвления.

Поставлены также задачи исследования. В частности, задача разработки и исследования свойств формализма, исходя из самых общих представлений о концептах, используемых для построения алгоритма, решающего некоторую задачу. В работе вводятся понятия предельной частично транзакционной памяти, предельной непараллельной объектно-транзакционной модели.

Рассмотрена проблема представимости параллельных и непараллельных алгоритмов с помощью объектно-транзакционных моделей, базирующихся на идее предельной частично транзакционной памяти, содержащей единственную транзакционную ячейку со специальным согласованием, в условиях наличия единственной алгоритмической конструкции – ветвления. По мнению авторов, впервые сформулировано понятие таких моделей, описаны основные принципы их функционирования, сформулированы их основные свойства. Сказанное позволяет сделать вывод о наличии новизны исследования.

В работе авторами сформулированы и представлены результаты исследования. В частности, доказаны теоремы о разрешимости такими моделями произвольных последовательных алгоритмов, вычислимых по Тьюрингу, а также о разрешимости произвольных алгоритмов с К параллельными ветвями, вычислимых системой из К параллельно работающих машин Тьюринга, взаимодействующих через общую ленту.

Проведенный анализ статьи также показал, что при подготовке статьи использованы различные источники и научная литература. Однако, список использованных источников и литературы достаточно скромен. Библиографический список составляет 6 источников, среди которых представлены научные публикации зарубежных и отечественных исследователей. В этой связи, полагаем, что источниковая база исследования не позволила авторам статьи определить уровень разработанности проблемы в отечественной науке. В статье также, по нашему мнению, не удалось развернуть полноценную научную дискуссию. Однако данное обстоятельство, в целом не влияет на ее научность, глубину исследовательской концепции и научные результаты, отраженные в статье.

Статья структурирована. Положительно следует оценить наличие методологической базы исследования, подходов. В частности, в данной работе используется авторский вариант – предельная частично транзакционная память, включающая единственную специальную транзакционную ячейку.

Статья интересная. В статье представлены формулы, теоремы и их доказательства. Статья способна вызвать читательский интерес.

Считаем, что рецензируемая научная статья на тему: «Объектно-транзакционные модели программ на алгоритмических языках» соответствует необходимым требованиям, предъявляемым к такому виду научных работ. Она способна вызвать читательский интерес и рекомендуется к опубликованию в искомом научном журнале.