

Math-Net.Ru

All Russian mathematical portal

N. N. Nepeivoda, Using overflows to control parallel and distributed computations,
Program Systems: Theory and Applications, 2017,
Volume 8, Issue 3, 87–107

<https://www.mathnet.ru/eng/ps268>

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use
<https://www.mathnet.ru/eng/agreement>

Download details:

IP: 18.97.9.175

May 20, 2025, 09:10:07



Н. Н. Непейвода

Использование локализации и переполнения для управления параллельными и распределёнными вычислениями

Аннотация. Применительно к задачам суперкомпьютинга и сверхбольших баз данных рассматривается абстрактная концепция переполнения и показывается, как её использовать для организации и оптимизации распределённых и параллельных вычислений. Работа является второй в серии, посвящённой методам организации вычислений над локальными системами.

Ключевые слова и фразы: локальность, расслоение вычислений, расслоение данных, переполнение.

Статья является продолжением серии, начатой работой [1]. Определения ограниченности, локальности, прямоточности, введённые в данной работе, используются здесь без модификации.

1. Практические и содержательные мотивировки

Ассоциативные и коммутативные операции играют важнейшую роль при распределённых вычислениях. В частности, в системах типа MAPREDUCE [2] они служат инструментом «горизонтального» распараллеливания вычислений, когда частичные результаты постепенно собираются на многих независимо и асинхронно работающих процессорах. В данной статье предлагается подход, дополняющий горизонтальное распараллеливание бинарной операции вертикальным, когда на процессорах обрабатываются частичные данные, а соединение в общую информационную структуру производится после конца распределённых вычислений. Это может явиться ещё одним фактором, повышающим безопасность распределённых вычислений: каждый обработчик, кроме двух доверенных в начале и в конце обработки данных, знает лишь частичную информацию об обрабатываемой структуре.

Проект проводится при финансовой поддержке РАН, проект 46П «Сверхвысокопроизводительные базы данных».

© Н. Н. Непейвода, 2017

© ИНСТИТУТ ПРОГРАММНЫХ СИСТЕМ ИМЕНИ А. К. АЙЛАМАЗЯНА РАН, 2017

© ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ, 2017

DOI: 10.25209/2079-3316-2017-8-3-87-107

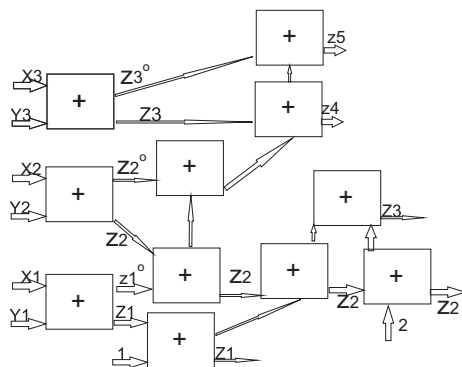


Рис. 1. Пример сети прямоточных вычислений (комбинатор)

Рассматриваемый подход важен для обратимых вычислений. Как заметил Ландауэр [3], они должны быть «баллистическими», что означает: *пути вычислений однозначно определены при задании исходных данных*. Для физической реализации часто необходимо второе условие: отсутствие запоминания промежуточных данных, сразу же направляемых на следующий обрабатывающий узел. В этом отношении баллистические вычисления схожи с аналоговыми. Такое требование полезно и для распределённых вычислений в сверхбольших базах данных (СВБД), чтобы не было нужды неизвестно сколько хранить частичные данные.

ПРИМЕР 1. *Дополнительной мотивировкой данного исследования явилось то, что структура комбинатора в рис. 1, созданная для реализации частного вида прямоточных оптических вычислений, оказалась работающей и для аналоговых, и для организации распределённых вычислений. Значит, за ней стоит общее теоретическое понятие.*

ПРИМЕР 2. *Переход к прямоточным вычислениям часто требует точного подбора преобразователей и представления данных. Показателен в этом отношении пример, найденный Червяковым и др. [4, глава 2.3]. Известно, что системы остаточных классов позволяют эффективно реализовывать многие операции, но некоторые, элементарные в позиционных системах, трудны. В частности, таковы операции перевода между системой остаточных классов и позиционной. Но при подходящем подборе представления и элементов для реализации (нейросеть) её можно реализовать как прямоточные*

вычисления (нейронная сеть прямого распространения, в терминах школы Галушкина). В частности, показано, что для преобразования 64-разрядных чисел в систему остаточных классов по модулям

5, 7, 11, 13, 17, 19, 23, 29, 37, 41, 43, 47, 53, 59, 61

достаточна четырёхслойная прямооточная нейронная сеть и не требуется синхронизации.

2. Алгебраическая и логическая техника

Для формализации локализации вычислений нам понадобится алгебраический базис. Действия внутри локального пространства данных могут вывести нас за пределы данной области. Результат может оказаться в некоторой области переполнения. Простейшим примером является сложение цифр в обычных системах счисления. При этом может получиться число из двух разрядов. Множеством переполнения для десятичных цифр являются числа 10–18. Повторное действие в множестве переполнения может вывести за его пределы, поэтому целесообразно в таких случаях считать результатом ошибку.

Структуры переполнения возникают часто и в самых разных областях. Поэтому целесообразно дать им чисто алгебраическое определение с помощью понятий, модифицирующих общие структуры из книги Мальцева [5]. Напомним, что в алгебраической системе первого порядка может быть несколько исходных множеств (примитивных типов данных), констант, предикатов и функциональных символов. В случае, если система одноосновная (с одним исходным типом данных), типы данных не указываются. Если явно не указано противное, в данной работе алгебраическая система первопорядковая и одноосновная, для всех элементов исходных типов имеются константы. У Мальцева и в других алгебраических работах рассматриваются структуры, где все операции всюду определены. В работах по информатике, наоборот, в последние десятилетия, особенно после работ Д. Скотта [6], стало принято рассматривать ошибки и частично определённые функции как типичный случай.

Алгебраизируем частичность и ошибочность, воспользовавшись заодно результатами современной многозначной логики. Определение сигнатуры оставим классическим, поэтому повторять его не будем.

Напомним, что слабая трёхзначная логика Клини задаётся добавлением к стандартным логическим значениями {**истина t**, **ложь f**} значения **ошибка u**. Любая логическая операция, один из аргументов которой есть **u**, даёт **u**. Семантику кванторов определим отдельно. Элемент «ошибка» будем обозначать \perp , как принято в информатике. Множество $X \cup \{\perp\}$ обозначается X_{\perp} .

ОПРЕДЕЛЕНИЕ 1. Пусть задана сигнатура σ . Частично определённая алгебраическая система сигнатуры σ — непустое множество, называемое носителем U , не содержащее \perp , и функция означивания \mathfrak{z} на сигнатуре σ , задающая:

- (а) для каждой константы c элемент $\mathfrak{z}[c] \in U$;
- (б) для каждого n -местного функционального символа f функцию $\mathfrak{z}[f] \in (U_{\perp})^n \rightarrow (U_{\perp})$, сохраняющую \perp : если один из аргументов \perp , то и значение функции \perp ;
- (в) для каждого n -местного предикатного символа P функцию $\mathfrak{z}[P] \in (U_{\perp})^n \rightarrow \{t, f, u\}$, сильно сохраняющую \perp : $\mathfrak{z}[P](a_1, \dots, a_n) = u$ тогда и только тогда, когда хотя бы один из аргументов $a_i = \perp$.

Таким образом, мы принимаем радикальное решение: если встречается ошибка, то значение любого выражения над ней ошибка. Предикаты дают третье значение лишь на ошибочных аргументах. Тем самым верна

ЛЕММА 1. *Замкнутая пропозициональная формула, в которой значения всех термов не равны \perp , не может принимать значение u .*

ДОКАЗАТЕЛЬСТВО. В этом случае значения всех элементарных подформул принадлежат $\{t, f\}$, и таблицы истинности выдают значения из того же множества. \square

Введём следующее требование, не ограничивающее общности (любую частично определённую алгебраическую систему можно пополнить константами для его выполнения), позволяющее ввести удобные сокращения.

Требование 1. Любой элемент U является значением замкнутого терма сигнатуры σ .

Теперь можно говорить вместо «функциональный (предикатный) символ» просто функция или предикат, и отождествлять термы с их значениями.

ОПРЕДЕЛЕНИЕ 2. $\mathfrak{R} \subseteq U$ — регулярное подмножество, если значения всех функций на элементах \mathfrak{R} принадлежат U . Его элементы называются регулярными значениями.

Заметим: не требуется, чтобы результат функции над регулярными значениями был регулярным. Говорить о регулярности элемента безотносительно к \mathfrak{R} нельзя, поскольку на кортежах значений, лишь часть из которых принадлежит \mathfrak{R} , функции могут давать ошибку. Какой-то смысл имеет регулярность элемента лишь для одноместных функций.

ПРИМЕР 3. Рассмотрим множество квадратных матриц некоторой размерности над полем действительных чисел \mathbb{R} с операциями сложения, вычитания, умножения и взятия обратной. Тогда множество невырожденных матриц можно считать множеством регулярных элементов, но, в частности, операция сложения может вывести за пределы этого множества.

Эта алгебраическая система помогает увидеть существенное свойство: единичная матрица сохраняет свойства и для нерегулярных элементов; алгебраические тождества над регулярными элементами остаются справедливыми и для нерегулярных, если результаты действий определены.

Введём важные семантические различия для предварённых формул. Как обычно, последовательности одинаковых кванторов будем сокращать в один кванторный префикс с несколькими переменными.

ОПРЕДЕЛЕНИЕ 3. Пусть \mathbb{K} — квантор, $\mathbb{K}x_1, \dots, x_n \mathcal{A}$ — замкнутая формула. Тогда её подстановочным примером называется результат подстановки замкнутых термов t_1, \dots, t_n в \mathcal{A} .

Ключевой модификацией семантических понятий для наших целей, когда имеется выделенное множество регулярных элементов и возможны переполнения, является оценка формулы на подмножестве U . Получить её с помощью ограниченных кванторов в слабой логике Клини не удаётся, и приходится вводить отдельно. Дадим следующие семантические определения индукцией по сложности кванторных префиксов предварённых формул.

ОПРЕДЕЛЕНИЕ 4 (Справедливость, истинность и ложность замкнутых предварённых формул на непустом $X \subseteq U$).

- (а) Бескванторная формула истинна на X , когда она имеет значение t . Она справедлива, когда имеет значение t или u .
- (б) Формула $\forall x_1, \dots, x_n \mathcal{A}$ истинна на X , когда любой её подстановочный пример со значениями термов $t_i \in X$ истинен на X . Она справедлива на X , когда любой её подстановочный пример со значениями термов $t_i \in X$ справедлив на X . Она ложна на X , когда некоторый её подстановочный пример со значениями термов $t_i \in X$ ложен на X , а остальные имеют значения из $\{t, f\}$.

(в) Формула $\exists x_1, \dots, x_n \mathcal{A}$ истинна на X , когда некоторый её подстановочный пример со значениями термов $t_i \in X$ истинен на X и ни один подстановочный пример со значениями термов $t_i \in X$ не имеет значение \mathbf{u} . Она справедлива на X , когда некоторый её подстановочный пример со значениями термов $t_i \in X$ истинен на X , а остальные имеют значения из $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. Она ложна на X , когда любой её подстановочный пример со значениями термов $t_i \in X$ ложен на X .

Считается, что истинные предварённые формулы имеют на X значение \mathbf{t} , ложные — \mathbf{f} , справедливые, но не истинные — \mathbf{u} .

В последующем тексте данной главы все логические формулы являются предварёнными, если явно не оговорено противное. Итак, справедливость допускает ошибочные значения части примеров, а истинность требует отсутствия ошибок. Заметим, что ложные значения подстановочных примеров могут разрушить справедливость быстрее, чем ошибки. Тождественно ошибочная формула справедлива, если она является Π -формулой $\forall x_1, \dots, x_n \mathcal{A}$, где \mathcal{A} бескванторная.

ЛЕММА 2. Если формула истинна на X , то она справедлива на X .

ДОКАЗАТЕЛЬСТВО. Непосредственно следует из определения. \square

ПРИМЕР 4. Рассмотрим теорию действительных чисел с операциями $a + b$, $a \cdot b$, взятием противоположного $-a$ и обратного a^{-1} . Множество регулярных элементов — все ненулевые, так что единственный элемент переполнения 0 . Тогда формула $\forall x (x \neq -x \Rightarrow x \cdot x^{-1} = 1)$ истинна на множестве регулярных элементов, но всего лишь справедлива на всём \mathbb{R} . Посылка $x \neq -x$ понадобилась, чтобы избежать упоминания 0 в утверждениях о регулярных элементах. Логически следующая из неё формула

$$\forall x \exists y (x \neq -x \Rightarrow x \cdot y = 1)$$

истинна на множестве регулярных элементов, но даже не справедлива на всём \mathbb{R} .

Формула $\exists x (x + (-x))^{-1} = (x + (-x))^{-1}$ не имеет истинностного значения. Формула $\forall x (x + (-x))^{-1} = (x + (-x))^{-1}$ справедлива.

ЛЕММА 3. Пусть X — регулярное множество. Замкнутая формула, в которой все термы являются переменными, константами из X либо имеют вид $f(x_1, \dots, x_n)$, где x_i — переменные либо константы из X , истинна на X тогда и только тогда, когда она справедлива на нём.

ДОКАЗАТЕЛЬСТВО. Следствие леммы 1. □

ОПРЕДЕЛЕНИЕ 5. Пусть выделено регулярное подмножество \mathfrak{A} . Будем говорить, что теория Th , аксиомы которой являются предварёнными формулами (в дальнейшем просто теория), справедлива на \mathfrak{A} , если все её аксиомы истинны на \mathfrak{A} и справедливы на U .

ОПРЕДЕЛЕНИЕ 6. Алгебраическая система с переполнением (АСП) это частично определённая алгебраическая система, в которой выделено подмножество регулярных значений \mathfrak{A} и задана справедливая на \mathfrak{A} теория Th . В этом случае множество $U \setminus \mathfrak{A}$ обозначается через \mathfrak{D} и называется областью переполнений, а его элементы (полными) переполнениями.

Таким образом, свойства ассоциативности и коммутативности, нуля и единицы сохраняются и при переполнении, если промежуточные результаты не выходят за область определённости.

ЛЕММА 4. Если формула \mathfrak{A} справедлива на каждой АСП, на которой справедлива теория Th , то она выводима в Th при помощи классической логики предикатов.

ДОКАЗАТЕЛЬСТВО. Следствие теоремы о полноте классической логики. Если \mathfrak{A} не выводима в Th , то можно построить алгебраическую систему, в которой истинны все аксиомы Th и ложна \mathfrak{A} . Эта модель превращается в АСП заданием $\mathfrak{A} = U$. □

ОПРЕДЕЛЕНИЕ 7. Формулы $\mathfrak{A}(x_1, \dots, x_n)$ и $\mathfrak{B}(x_1, \dots, x_n)$ равнозначны на X , если при каждом $a_1, \dots, a_n \in X$ $\mathfrak{A}(a_1, \dots, a_n)$ и $\mathfrak{B}(a_1, \dots, a_n)$ одновременно имеют либо не имеют значения и в случае определённости их значения совпадают.

ОПРЕДЕЛЕНИЕ 8. Пропозициональной матрицей формулы называется её бескванторная часть. Областью переменной называется минимальная подформула, содержащая все её вхождения. Переменные x, y независимы, если их области не пересекаются.

Рассмотрим теперь проблему инвариантности семантических понятий относительно конкретного варианта предварённой формы. Очевидно, что нетривиальные преобразования (в частности, эквивалентные с точки зрения классической логики изменения пропозициональной матрицы) могут повлиять на значение формулы в нашей многозначной логике. Поэтому выделим класс преобразований, которые можно рассматривать как результат изменения порядка выноса независимых кванторов в кванторный префикс.

ОПРЕДЕЛЕНИЕ 9. Однородные кванторы — *последовательность одинаковых кванторов с разными переменными. Кванторный префикс формулы является последовательностью однородных кванторов, в которой \forall и \exists чередуются. Переменная x впереди переменной y , если она находится в однородном кванторе, предшествующем однородному квантору, в котором находится y . Две формулы \mathfrak{A} , \mathfrak{B} называются несущественно отличающимися, если они могут быть преобразованы друг в друга последовательностью следующих шагов:*

Тривиальная перестановка: обмен местами двух переменных в одном и том же однородном кванторе.

Существенная перестановка: перестановка двух рядом расположенных независимых переменных из разных однородных кванторов вместе с их кванторами при условии, что области переменных, связанных всеми последующими кванторами в кванторном префиксе, либо вложены в область одной из переставляемых переменных, либо не пересекаются с обеими.

ПРИМЕР 5. *Формулы*

$$\forall x \forall y \exists z \forall u \exists v (A(x, z) \& B(x, u) \& C(x, y) \& D(y, v)),$$

$$\forall u \forall y \forall x \exists z \exists v (A(x, z) \& B(x, u) \& C(x, y) \& D(y, v)),$$

$$\forall x \forall y \exists v \exists z \forall u (A(x, z) \& B(x, u) \& C(x, y) \& D(y, v))$$

отличаются несущественно.

Для того, чтобы обосновать сохранение значения несущественно отличающихся формул, установим следующее свойство независимых переменных.

ОПРЕДЕЛЕНИЕ 10. *Формула $\mathfrak{A}(x, y)$ обладает свойством униформизации по переменным x, y на множестве X , если для каждого логического значения Γ найдётся такое $b_\Gamma \in X$, что, если при некоторых $a, b \in X$ $\mathfrak{A}(a, b)$ имеет значение Γ , то $\mathfrak{A}(a, b_\Gamma)$ имеет значение Γ .*

ЛЕММА 5. *Если x, y — единственные свободные переменные формулы $\mathfrak{A}(x, y)$, и она обладает по ним свойством униформизации, то на любом множестве X в любой частично определённой алгебраической системе формулы $\forall x \exists y \mathfrak{A}(x, y)$ и $\exists y \forall x \mathfrak{A}(x, y)$ равнозначны.*

ДОКАЗАТЕЛЬСТВО. Если формула $\forall x \exists y \mathfrak{A}(x, y)$ имеет значение \mathfrak{t} , то по свойству униформизации найдётся такое $b \in X$, что при любом $a \in X$ $\mathfrak{A}(a, b)$ истинна, и по определению истинности формула $\mathfrak{A}(x, y)$ принимает лишь значения $\{\mathfrak{t}, \mathfrak{f}\}$. Следовательно, истинна $\exists y \forall x \mathfrak{A}(x, y)$. В обратную сторону следствие тривиально.

Если формула $\exists y \forall x \mathfrak{A}(x, y)$ имеет значение \mathfrak{f} , то по свойству униформизации найдётся такое $a \in X$, что при любом $b \in X$ $\mathfrak{A}(a, b)$ ложна, и по определению ложности формула $\mathfrak{A}(x, y)$ принимает лишь значения $\{\mathfrak{t}, \mathfrak{f}\}$. Следовательно, ложна $\forall x \exists y \mathfrak{A}(x, y)$. В обратную сторону следствие тривиально.

Если формула $\forall x \exists y \mathfrak{A}(x, y)$ имеет значение \mathfrak{u} , то по определению справедливости для каждого $a \in X$ найдётся $b \in X$, при котором $\mathfrak{A}(a, b)$ истинна, и применяя свойство униформизации, находим $b_0 \in X$, при котором $\forall x \mathfrak{A}(x, b_0)$ истинна. А поскольку формула справедлива, найдутся такие $a_1, b_1 \in X$, что значение $\mathfrak{A}(a_1, b_1)$ есть \mathfrak{u} . Таким образом, $\exists y \forall x \mathfrak{A}(x, y)$ также имеет значение \mathfrak{u} . В обратную сторону доказательство тривиально.

Оставшийся случай — когда одна из формул не имеет значения. По правилу Шерлока Холмса (единственной оставшейся альтернативы), тогда и вторая его не имеет. \square

ЛЕММА 6. *Если x, y — единственные свободные переменные пропозициональной формулы $\mathfrak{A}(x, y)$, и они независимы, то на любом множестве X в любой частично определённой алгебраической системе $\mathfrak{A}(x, y)$ обладает свойством униформизации по x, y и по y, x .*

ДОКАЗАТЕЛЬСТВО. Доказательство ведётся индукцией по построению формулы $\mathfrak{A}(x, y)$.

Базис индукции. Пусть формула $\mathfrak{A}(x, y)$ пропозициональная и получается применением бинарной связки к формулам $\mathfrak{B}(x)$ и $\mathfrak{C}(y)$. Достаточно рассмотреть $\&$ и \vee , поскольку \Rightarrow представляется в виде $\neg A \vee B$, и за одну из формул можно взять $\neg A$.

Б1. Если $\mathfrak{B}(a) \& \mathfrak{C}(b)$ истинна, то обе подформулы истинны, и если истинна $\mathfrak{B}(a_1) \& \mathfrak{C}(b_1)$, то истинна также $\mathfrak{B}(a_1) \& \mathfrak{C}(b)$, и в качестве искомого $b_{\mathfrak{t}}$ достаточно взять b .

Если $\mathfrak{B}(a) \vee \mathfrak{C}(b)$ истинна, то одна из подформул истинна, и если такова $\mathfrak{C}(b)$, то в качестве искомого b_t достаточно взять b . Если же $\mathfrak{C}(b)$ ложна, то просматриваем все истинные формулы $\mathfrak{B}(a_1) \vee \mathfrak{C}(b_1)$, и если найдётся истинная $\mathfrak{C}(b_1)$, то в качестве искомого b_t достаточно взять b_1 . Если же таковой нет, то в таких случаях всегда истинна $\mathfrak{B}(a_1)$, и в качестве искомого b_t достаточно взять b .

Б2. Если $\mathfrak{B}(a) \& \mathfrak{C}(b)$ справедлива, то одна из подформул справедлива, и если и если такова $\mathfrak{C}(b)$, то в качестве искомого b_u достаточно взять b . Если же $\mathfrak{C}(b)$ истинна либо ложна, то просматриваем все справедливые формулы $\mathfrak{B}(a_1) \& \mathfrak{C}(b_1)$, и если найдётся справедливая $\mathfrak{C}(b_1)$, то в качестве искомого b_u достаточно взять b_1 . Если же таковой нет, то в таких случаях всегда справедлива $\mathfrak{B}(a_1)$, и в качестве искомого b_t достаточно взять b . Случай \vee разбирается совершенно так же.

Б3. Случай ложности разбирается двойственно к случаю истинности.

Б4. Для пропозициональных формул значение всегда существует, и четвёртый случай отбрасывается.

Б5. Поскольку понятие независимости симметрично, случай перестановки переменных отдельного рассмотрения не заслуживает. \square

Заметим, что данная лемма существенно зависит от выбора исходных связей логической системы. В частности, если добавить связку \equiv , она перестаёт быть верной.

ЛЕММА 7. *Если*

(1)

$$\mathbf{K}_1 x_1 \dots \mathbf{K}_n x_n \forall x \exists y \mathbf{K}_{n+1} z_1 \dots \mathbf{K}_{n+k} z_k \mathfrak{A}(x_1, \dots, x_n, x, y, z_1, \dots, z_k)$$

предварённая формула, в пропозициональной матрице которой x, y и все z_i, x независимые переменные, то она равнозначна с

(2)

$$\mathbf{K}_1 x_1 \dots \mathbf{K}_n x_n \exists y \forall x \mathbf{K}_{n+1} z_1 \dots \mathbf{K}_{n+k} z_k \mathfrak{A}(x_1, \dots, x_n, x, y, z_1, \dots, z_k).$$

ДОКАЗАТЕЛЬСТВО. Чтобы доказать равнозначность формул (1) и (2), достаточно установить равнозначность всех пар формул

$$(3) \quad \forall x \exists y \mathbf{K}_{n+1} z_1 \dots \mathbf{K}_{n+k} z_k \mathfrak{A}(t_1, \dots, t_n, x, y, z_1, \dots, z_k)$$

и

$$(4) \quad \exists y \forall x \mathbf{K}_{n+1} z_1 \dots \mathbf{K}_{n+k} z_k \mathfrak{A}(t_1, \dots, t_n, x, y, z_1, \dots, z_k),$$

где $t_1, \dots, t_n \in X$.

Будем доказывать более сильное утверждение, что в дополнение к сказанному во всех

$$\mathbf{K}_{n+1}z_1 \dots \mathbf{K}_{n+k}z_k \mathfrak{A}(t_1, \dots, t_n, x, y, z_1, \dots, z_k)$$

x, y обладают свойством униформизации. Доказательство ведётся индукцией по длине кванторного префикса $\mathbf{K}_{n+1}z_1 \dots \mathbf{K}_{n+k}z_k$.

Базис индукции. Если кванторный префикс отсутствует, то равнозначность и униформизация следуют из лемм 6, 5.

Допущение индукции. Пусть доказана равнозначность всех пар формул

$$(5) \quad \forall x \exists y \mathfrak{A}(t_1, \dots, t_n, x, y, s)$$

и

$$(6) \quad \exists y \forall x \mathfrak{A}(t_1, \dots, t_n, x, y, s),$$

где $t_1, \dots, t_n, s \in X$ и в формуле $\mathfrak{A}(x_1, \dots, x_n, x, y, z)$ переменные x, y и x, z независимы, x, y и x, z обладают свойством униформизации.

Рассмотрим формулу (5) и разберём случаи её возможных значений.

1. **Истина.** Тогда при каждом $a \in X$ существуют $b, c \in X$, такое, что $\mathfrak{A}(t_1, \dots, t_n, a, b, c)$ истинна. По предположению индукции (униформизация) тогда есть такие b, c , единые для всех a , и истинна $\exists y \forall x \exists z \mathfrak{A}(t_1, \dots, t_n, x, y, z)$
2. **Справедлива.** Доказательство совершенно аналогично.
3. **Ложна.** Тогда по правилам классической логики ложна и формула $\exists y \forall x \exists z \mathfrak{A}(t_1, \dots, t_n, x, y, z)$.
4. **Нет значения.** Тогда возможны следующие случаи.
 - 4.1. При некоторых $a, b \in X \exists z \mathfrak{A}(t_1, \dots, t_n, a, b, z)$ не имеет значения. Тогда и обе более сложные формулы не имеют значения.
 - 4.2. Случай 4.1. не выполнен, но при некотором $a \in X$ формула $\exists y \exists z \mathfrak{A}(t_1, \dots, t_n, a, y, z)$ не имеет значения. Тогда при любых $b \in X$ справедлива или ложна $\exists z \mathfrak{A}(t_1, \dots, t_n, a, b, z)$ и где-то выполнены оба этих случая. Но тогда по свойству униформизации по x, z , есть такое $c \in X$, что при всех $a \in X \mathfrak{A}(t_1, \dots, t_n, a, b, c)$ справедлива, и справедлива $\forall x \exists z \mathfrak{A}(t_1, \dots, t_n, x, b, z)$. Аналогично находим значение, при котором $\forall x \exists z \mathfrak{A}(t_1, \dots, t_n, x, b, z)$ ложна. При этом не может быть значения, при котором она истинна. Значит, по определению $\exists y \forall x \exists z \mathfrak{A}(t_1, \dots, t_n, x, y, z)$ не имеет значения.

5. Если (6) истинна, ложна или справедлива, то по определению логических значений такова же (5). И по правилу Шерлока Холмса тогда из того, что (6) не имеет значений, следует, что и (5) не имеет значения. \square

ТЕОРЕМА 1. *Если x, y — независимые в предварённой формуле $\mathfrak{A}(x, y)$ и единственные в ней свободные переменные, то на любом множестве X в любой частично определённой алгебраической системе формулы $\forall x \exists y \mathfrak{A}(x, y)$ и $\exists y \forall x \mathfrak{A}(x, y)$ равнозначны.*

Непосредственное следствие из предыдущих лемм.

Следствие. *Две несущественно отличающиеся формулы A, B равнозначны.*

Любая тривиальная или существенная перестановка покрывается предыдущими леммами и теоремой.

Итак, доказанные нами результаты обосновывают корректность нашей логической семантики по отношению к предварённым формулам и одновременно выявляют тонкие аспекты, которые традиционно игнорируются в работах, где используются не всюду определённые выражения и выражения, значения которых могут выходить за область надёжных вычислений.

В случае рассмотрения произвольных формул возникает четырёхзначная логика с двумя выделенными значениями, которая лежит несколько в стороне от основной цели нашей работы и поэтому здесь далее не рассматривается.

3. Локализация

3.1. Горизонтальное разбиение и соединение

Пусть имеется тип данных T и бинарная операция на нём \oplus , обладающая единичным элементом e и, возможно, нулевым элементом 0 . Пусть задано также конечное число n типов данных T_i , называемых локалами данных, являющихся системами с переполнением, на каждом из которых задана частичная операция \oplus_i , обладающая единицей (и нулём, если он есть у исходной). Элементы T_i называются локальными данными.

Пусть задана операция сборки *ASSEMBLE*, принимающая на вход n -ку элементов T_i и выдающая элемент T . Если

$$ASSEMBLE(\langle x_1, \dots, x_n \rangle) = X,$$

то кортеж $\langle x_1, \dots, x_n \rangle$ называется разбиением X . Эта операция сюръективна. Через $x \oplus^* y$ обозначим бинарную операцию на прямом произведении $\prod_{i=1}^n T_i$, определяемую как кортеж всех $x_i \oplus y_i$.

Поскольку понадобится вкладывать элементы T_i в T , обозначим $\langle x_i \rangle$ результат применения *ASSEMBLE* к вектору, i -тая компонента которого x_i , а остальные — единичные элементы соответствующих T_i . Обобщая, через $\langle \langle x_i \rangle_{i \in J} \rangle$ обозначим результат сборки вектора, все неединичные компоненты которого принадлежат подмножеству $J \subseteq [1 : n]$.

Заметим, что не накладывается требование единственности разбиения. Это важно для дальнейшего. Требуется лишь наличие односторонней обратной операции *SPLIT*(X), выдающей по данным одно из возможных их разбиений на кортежи из регулярных элементов.

$$(7) \quad ASSEMBLE(SPLIT(X)) = X.$$

Операции разбиения и \oplus согласованы друг с другом:

$$(8) \quad ASSEMBLE(SPLIT(x) \oplus^* SPLIT(y)) = x \oplus y,$$

то есть *SPLIT* является гомоморфизмом коммутативных полугрупп, если операции всюду определены.

ПРИМЕР 6. *Простейшим примером такого разбиения является разбиение чисел в позиционной системе на разряды для операции сложения. В случае системы счисления с избыточностью разбиение неоднозначно, но восстановление остаётся однозначным.*

ПРИМЕР 7. *Пусть заданы два векторных поля на замкнутой связной области векторного пространства Ξ и линейный коммутативный оператор над ними, обладающий свойством локальности: значения результата в точке зависят лишь от значений аргументов в некоторой окрестности этой точки. Простейшим примером такого оператора является*

$$(\alpha \oplus \beta)(\vec{x}) = \int \dots \int_{S(r(\vec{x}), \vec{x})} \alpha(\vec{y}) + \beta(\vec{y}) d\vec{y}.$$

Здесь $S(a, \vec{x})$ — сфера радиуса a с центром \vec{x} , $r(\vec{x})$ — непрерывный оператор из фазового пространства в \mathbb{R}^+ .

Разбиением для этой операции может служить любая система открытых подобластей ΞD_i , обладающая следующим свойством: любая точка \vec{x} содержится в одной из областей вместе со своей окрестностью $S(r(\vec{x}), \vec{x})$, каждое из подполей α_i обращается в ноль вне D_i и

$$\alpha(\vec{x}) = \sum_i \alpha_i(\vec{x}).$$

Такое разбиение назовём локализованным. Очевидно, что в локализованном разбиении области пересекаются, если их больше одной и пространство связно.

В этом случае регулярными элементами T_i целесообразно считать векторные поля, обращающиеся в ноль на точках x , окрестности $S(r(\vec{x}), \vec{x})$ которых не вложены в D_i . Но результат операции скорее всего будет переполнением, поскольку ненулевыми будут лишь значения вне D_i .

Разбить пространство на области можно разными способами, поскольку на пересечении областей можно выбрать любое аддитивное разложение, ограничиваемое с прикладной точки зрения лишь требованием, чтобы оно давало регулярные поля, достаточно гладкие, чтобы не нарушать корректности применяемых методов и (или) стабильности работы вычислительных элементов.

3.2. Вертикальная локализация

Рассмотрим пример, идейно базирующийся на 7, но показывающий необходимость ещё одного вида локализации.

ПРИМЕР 8. Пусть мы вычисляем непрерывный оператор над векторным полем, обладающий свойствами, близким к линейности:

$$F(a\vec{X} + \vec{Y}) = \varphi(a)F(\vec{X}) + F(\vec{Y}); \quad F(0) = 0.$$

Пусть в разных подобластях замкнутой связной области векторного пространства Ξ значения поля \vec{x} и значения оператора $f(\vec{x})$ имеют различный порядок, а наши вычислительные элементы могут работать лишь в ограниченном диапазоне изменений аргумента и результата. Пусть, подобно примеру 8, оператор обладает свойством локальности: значения результата в точке зависят лишь от значений аргументов в некоторой окрестности этой точки. Зададим локализованное разбиение таким образом, чтобы значения оператора в каждой подобласти D_i были ограничены по норме константой a_i . Обозначим через d_i верхнюю грань расстояний точек области D_i от начала координат. Выберем в каждой подобласти точку \vec{o}_i , расстояние от которой до любой точки области не больше $d_i/2$: центр области. Зададим преобразование векторного поля

$$\vec{X}_i(\vec{x}) = \vec{X}(0.5d_i \cdot \vec{x} + \vec{o}_i).$$

На каждой подобласти зададим оператор

$$F_i(\vec{X}_i)(\vec{x}) = F(\vec{X})(\vec{x})/a_i.$$

Это подобно операции масштабирования при аналоговых либо других вычислениях на физической основе. В этом случае при разбиении поля части значений поля, идущие в разные области, умножаются на константы. А при обратной сборке используется другая совокупность констант: массив значений $\varphi(a_i)$. В данном случае операцией сборки является $ASSEMBLE(\vec{x} + 2\vec{y})$.

Пример показывает, что при физической реализации часто важно локализовать вычисления не только по подобластям пространства (что зачастую делается и ныне), но и по диапазонам значений аргумента. Этот второй вид локализации менее изучен (хотя автору неизвестна также столь общая и абстрактная математическая формулировка локализации по горизонтали, как приведённая в данной работе; все известные ему примеры, в частности [7, 9], под неё подпадают.)

4. Управление параллельными и распределёнными вычислениями

В практических задачах переполнение, как правило, идёт в «непосредственно прилежащие» локалы. При горизонтальной локализации в соседние области пространства, при вертикальной — в соседние «ряды». Этим пользуются при организации параллельных и распределённых вычислений. Дадим общее описание данного метода.

ОПРЕДЕЛЕНИЕ 11. Граф переполнений для операции \oplus — это ориентированный граф на $[1 : n]$, где n — число локалов, обладающий следующими свойствами:

- (1) Для каждого i имеется петля с началом и концом в i .
- (2) Если x, y — элементы T_i , $z = x \oplus y$, то z можно представить в виде вектора $\langle (z_j)_{j \in \text{Con}(i)} \rangle$ (вектор нормализованных переполнений). При этом $\langle x \rangle \oplus \langle y \rangle = \langle (z_j)_{j \in \text{Con}(i)} \rangle$.

Здесь $\text{Con}(i)$ — множество вершин, в которые ведут дуги из i . z_i называется регулярной частью результата, переполнение z_j считается ассоциированным с дугой графа $\langle i, j \rangle$. Переполнение имеется, если хоть одно из $z_j \neq e$, $j \neq i$.

Таким образом, переполнение можно представить как совокупность регулярных данных из локалов, непосредственно связанных с исходным. Если переполнение $z_j = e$, $j \neq i$, оно считается несуществующим и игнорируется.



Рис. 2. Граф переполнений для сложения в позиционной системе

ПРИМЕР 9. Для сложения в позиционной системе граф переполнений на рис. 2 линейен и однонаправлен. Для разбиения линейного оператора на подобласти граф переполнений связывает дугами области, пересекающиеся друг с другом, и его можно рассматривать как неориентированный.

В случае коммутативной ассоциативной операции изложенных требований достаточно для корректности операций с переполнением и организации процесса вычислений.

Алгоритм организации вычислительного процесса

Пусть каждый локал вычисляется отдельным «процессором», а операция ассоциативна и коммутативна.

- (А) Вычислить результат операции на каждом локале независимо.
- (Б) Преобразовать его в вектор нормализованных переполнений и послать каждое существующее переполнение в соответствующий процессор.
- (В) По мере получения переполнений от других, каждый процессор обрабатывает их в произвольном порядке, применяя команды (А) и (Б). Необработанные переполнения хранятся в очереди.
- (Г) Процесс завершается, когда не остаётся имеющихся переполнений.

Конец алгоритма

Этот алгоритм в каждом конкретном случае расслоения требует анализа на завершаемость и число итераций обработки переполнений. Легче всего анализ и организация вычислений осуществляются, если граф переполнений не содержит циклов, кроме петель. Далее, «произвольный порядок» означает, что в каждом конкретном случае он может и должен быть конкретизирован для достижения лучших характеристик вычислений.

ПРИМЕР 10. В случае обычной позиционной системы при сложении переполнения могут обрабатываться k шагов, где k — число разрядов. В случае системы с избыточностью для целых чисел можно достичь параллельного сложения за два шага, а для действительных за три [8, 10].

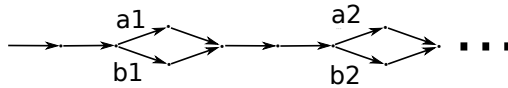


Рис. 3. Граф переполнений для схемы комбинатора

Если граф переполнений без циклов, в общем случае число итераций не меньше (максимального числа переполнений на пути без петель)+1. В случае графа (2) это число вершин. Поэтому целесообразно рассмотреть методы оптимизации обработки переполнений.

ПРИМЕР 11. В частности, в большинстве реальных приложений переполнения «малы» по сравнению с регулярными элементами, и несколько операций над переполнениями могут давать регулярный результат. Например, в десятичной системе при сложении переполнение равно 1, и девять переполнений можно сложить без дополнительного переполнения. Таким образом, простейшая стратегия оптимизации количества итераций: вначале действовать с переполнениями, и лишь затем корректировать регулярную часть.

5. Корректность комбинатора

В качестве приложения данных теоретических рассмотрений рассмотрим условия корректности схемы на рис. 1 для общего случая. Непосредственным анализом схемы выделяются три условия:

- (А) Операция ассоциативна и коммутативна.
- (Б) Регулярная часть операции, имеющей переполнение, даёт регулярный результат при добавлении одного переполнения.
- (В) Граф переполнений показан на рис. 3 (петли опускаем).

При этом переполнения можно организовать так, что по каждой паре дуг a_i , b_i идёт максимум одно переполнение.

Таким образом, мы установили, что схема почти не зависит от конкретного представления данных, лишь бы каждый третий слой «дублировал» следующий. Заметим, что требование Б выполняется для обычных позиционных систем счисления: для десятичной, например, максимум при переполнении 18, $8 + 1 = 9$. В любом таком случае организация передачи данных между «процессорами», выполняющими операцию, может быть прямоточной: данные без запоминания в промежуточных хранилищах передаются для обработки в следующей «процессор».

6. Дополнительные замечания

Данная работа показывает, как могут быть конкретизированы во многих случаях реальных вычислений понятия локальности и ограниченности из [1]. Рассмотрим некоторые возможности развития и приложений рассматриваемой теории.

Сразу видно, что работа в потенциале обогащает возможности систем типа MAPREDUCE [2]. Они соответствуют работе без переполнений, а здесь появляется возможность организовать частичную обработку, а затем доработку результатов.

Возникает задача дисциплины передачи и обработки переполнений, которая должна решаться в каждом отдельном случае применительно к его особенностям. Частично мы её затронули в примере 11 и в §5, когда было поставлено условие нетривиальности лишь одного переполнения из двух.

В случае графа переполнений с циклами условием конечности итераций является идемпотентность переполнений: любая последовательность операций над переполнениями за конечное число шагов сходится к e . Ввиду неточности реальных вычислений, условием конца может быть достаточная малость всех переполнений по некоторой норме. Видно, что данный алгоритм асинхронен и может конкретизироваться в данной вычислительной обстановке для достижения быстрой работы каждого «процессора». Условие конца глобально, потому что отдельный «процессор» может некоторое время не получать нетривиальных переполнений, а затем получить их.

Важным понятием при обработке переполнений является *влияние* переполнений данной вершины: множество вершин, результат в которых может измениться при изменении обрабатываемых значений в данной вершине (можно рассматривать как частный случай понятия влияния из [1]). Заметим, что влияние зависит не только от графа, но и от способа разбиения результата на регулярную часть и переполнения.

Ситуация, если операция с переполнениями некоммутативна или неассоциативна, видимо, может решаться лишь в частных случаях. Обратим внимание на один из возможных. Часто переполнения обрабатываются другой операцией, и она может быть ассоциативна и коммутативна. Скажем, переполнения при умножении октав Кэли [11] (некоммутативно и неассоциативно) могут обрабатываться при помощи сложения векторов действительных чисел.

Касательно терминологии. При горизонтальном разбиении локалы имеют сходство с клетками многообразий в топологии, при вертикальном — со слоями в расслоённом пространстве. Но в каждом случае есть и существенные отличия. Поэтому использован новый термин.

Автор благодарен рецензенту за многочисленные предложения по улучшению изложения.

Список литературы

- [1] Н. Н. Непейвода. «О некоторых возможностях локальных вычислений в теории систем и базах данных», *Программные системы: теория и приложения*, **7:4(31)** (2016), с. 145–160, URL: http://psta.psisiras.ru/read/psta2016_4_145-160.pdf ↑^{87,104}
- [2] J. Dean, S. Ghemawat. “MapReduce: simplified data processing on large clusters”, OSDI’04: Sixth Symposium on Operating System Design and Implementation (San Francisco, CA, December, 2004), pp. 13. ↑^{87,104}
- [3] Рольф Ландауэр. «Необратимость и выделение тепла в процессе вычислений», *Квантовый компьютер и квантовые вычисления. Том 2*, 1999, с. 9–32, ISBN: ISBN 5-7029-0338-2. ↑⁸⁸
- [4] Н. И. Червяков, П. А. Сахнюк, А. В. Шапошников, А. Н. Макоха, *Нейрокомпьютеры в остаточных классах, Нейрокомпьютеры и их применение*, т. **11**, Радиотехника, М., 2003, 270 с. ↑⁸⁸
- [5] А. И. Мальцев. *Алгебраические системы*, Наука, М., 1970, 392 с. ↑⁸⁹
- [6] Christopher Strachey. *Toward a mathematical semantics for computer languages Oxford Programming Research Group Technical Monograph. PRG-6*, Oxford, 1971. ↑⁸⁹
- [7] И. М. Соболев, С. В. Картышов, И. А. Кульчицкая, Ю. Л. Левитан. «О многокритериальной оптимизации математических моделей», *Математическое моделирование*, **6:6** (1994), с. 85–93. ↑¹⁰¹
- [8] Н. Н. Непейвода, И. Н. Григорьевский, Е. П. Лилитко. «О представлении действительных чисел», *Программные системы: теория и приложения*, **5:4** (2014), с. 105–121, URL: http://psta.psisiras.ru/read/psta2014_4_105-121.pdf ↑¹⁰²
- [9] А. Е. Бондарев. «Локализация пространственно-временных структур в пространстве определяющих параметров с применением параллельных вычислений», *Известия ЮФУ. Технические науки*, 2012, №6, с. 100–105. ↑¹⁰¹
- [10] А. Б. Шворин. «Параллельное сложение вещественных чисел в системах счисления с перекрытием», *Программные системы: теория и приложения*, **6:2** (2015), с. 101–117, URL: http://psta.psisiras.ru/read/psta2015_2_101-117.pdf ↑¹⁰²
- [11] Дж. С. Баэс. «Октонионы», *Гиперкомплексные числа в геометрии и физике*, **3:5-1** (2006), с. 120–176. ↑¹⁰⁴

Рекомендовал к публикации

д.ф.-м.н. С. В. Знаменский

Пример ссылки на эту публикацию:

Н. Н. Непейвода. «Использование локализации и переполнения для управления параллельными и распределёнными вычислениями», *Программные системы: теория и приложения*, 2017, 8:3(34), с. 87–107.

URL: http://psta.psiras.ru/read/psta2017_3_87-107.pdf

Об авторе:



Николай Николаевич Непейвода

Главный научный сотрудник ИПС РАН, научный руководитель работ. Более 200 публикаций по конструктивной математике, логике, информатике

e-mail:

nepejvodann@gmail.com

Nikolai Nepejvoda. *Using overflows to control parallel and distributed computations.*

ABSTRACT. An abstract algebraic notion of overflow is introduced and used for problems of parallel and distributed computing. Its applicability is illustrated by some examples. This work is second in the series devoted to memory-free computations and locality. (in Russian).

Key words and phrases: locality, parallel computing, distributed computing, overflow, layered data.

References

- [1] N. N. Nepejvoda. “Local computations in system theory and VLDB”, *Program Systems: Theory and Applications*, 7:4(31) (2016), pp. 145–160 (in Russian), URL: http://psta.psiras.ru/read/psta2016_4_145-160.pdf
- [2] J. Dean, S. Ghemawat. “MapReduce: simplified data processing on large clusters”, OSDI’04: Sixth Symposium on Operating System Design and Implementation (San Francisco, CA, December, 2004), pp. 13.
- [3] Rolf Landauer. “Irreversibility and heat generation in the computing process», *IBM Journal of Research and Development*, 5 (1961), pp. 183–191.
- [4] N. I. Chervyakov, P. A. Sakhnyuk, A. V. Shaposhnikov, A. N. Makokha, *Neurocomputers in residual classes*, Neurocomputers and their applications, vol. 11, Radiotekhnika, M., 2003 (in Russian), 270 p.

Supported by RAS (project 46P)..

© N. N. NEPEJVODA, 2017

© AILAMAZYAN PROGRAM SYSTEMS INSTITUTE OF RAS, 2017

© PROGRAM SYSTEMS: THEORY AND APPLICATIONS, 2017

DOI: 10.25209/2079-3316-2017-8-3-87-107

- [5] A. I. Mal'tsev. *Algebraicheskiye sistemy*, Nauka, M., 1970, 392 p.
- [6] Christopher Strachey. *Toward a mathematical semantics for computer languages Oxford Programming Research Group Technical Monograph. PRG-6*, Oxford, 1971.
- [7] I. M. Sobol', S. V. Kartyshov, I. A. Kul'chitskaya, Yu. L. Levitan. "Multicriteria optimization of the mathematical models", *Matematicheskoye modelirovaniye*, **6:6** (1994), pp. 85–93 (in Russian).
- [8] N. N. Nepeyvoda, I. N. Grigorevskiy, Ye. P. Lilitko. "New representation of real numbers", *Program Systems: Theory and Applications*, **5:4** (2014), pp. 105–121 (in Russian), URL: http://psta.psir.ru/read/psta2014_4_105-121.pdf
- [9] A. Ye. Bondarev. "Space-time structures localization in the space of character parameters by parallel computations", *Izvestiya YuFU. Tekhnicheskiye nauki*, 2012, no.6, pp. 100–105 (in Russian).
- [10] A. B. Shvorin. "Parallel addition of real numbers in overlaying numeration systems", *Program Systems: Theory and Applications*, **6:2** (2015), pp. 101–117 (in Russian), URL: http://psta.psir.ru/read/psta2015_2_101-117.pdf
- [11] Dzh. S. Baes. "The octonions", *Giperkompleksnyye chisla v geometrii i fizike*, **3:5-1** (2006), pp. 120–176 (in Russian).

Sample citation of this publication:

Nikolai Nepeyvoda. "Using overflows to control parallel and distributed computations", *Program systems: Theory and applications*, 2017, **8:3(34)**, pp. 87–107. (In Russian).

URL: http://psta.psir.ru/read/psta2017_3_87-107.pdf