

СВОЙСТВА НА УНИВЕРСАЛНИЯ АГРЕГАТОР $A \circ$

Ангел Тошков, Бургаски свободен университет e-mail: angel@bfu.bg

Абстракт: В статията се изследват фундаментални структурни свойства на универсалния математически агрегатор $A \circ$, въведен в [1], който обединява широк клас класически операции – сума, произведение, екстремуми, логически квантори, интеграли в дискретизиран вид, семирингови редукции и други, разглеждани като специални случаи на един общ оператор, дефиниран върху подходящ моноид. Представя се разширен формален анализ на основните алгебрични и функционални свойства на агрегатора, включително асоциативност, неутралност, комутативност (когато е приложима), инвариантност, монотонност и разделителност. Извеждат се необходимите условия за неговата коректност като оператор върху множества, а не само като списъчна редукция (fold). Дава се разширено описание на дефиницията на агрегатора, параметрите, участващите структури и мотивацията за изследване на тези свойства предвид бъдещи приложения в анализ, теория на изчисленията, оптимизация, обработка на данни, интегрални оператори и матрични редукции [1], [3], [7].

Ключови думи: универсален агрегатор, моноид, катаморфизъм, бинарна операция, редукция, агрегиране върху множества, семиринги, интегрални редукции.

PROPERTIES OF THE UNIVERSAL AGGREGATOR $A \circ$

Angel Toshkov, Burgas Free University e-mail: angel@bfu.bg

Abstract: This paper investigates the fundamental structural properties of the universal mathematical aggregator $A \circ$, introduced in [1], which unifies a wide class of classical operators—sum, product, extrema, logical quantifiers, discretized integrals, semiring reductions, and others—viewed as strict special instances of a single operator defined over an appropriate monoid. We provide an extended formal analysis of the essential algebraic and functional properties of the aggregator, including associativity, identity, commutativity (when applicable), invariance, monotonicity, and decomposability. The necessary conditions are derived for the aggregator to operate naturally over sets rather than merely as a list-based fold. A detailed explanation is given for all parameters, structural components, and operators entering the definition, together with the motivation for examining these properties in view of future applications in analysis, computation theory, optimization, data aggregation, integral operators, and matrix reductions [1], [3], [7].

Keywords: universal aggregator, monoid, catamorphism, binary operation, reduction, set-based aggregation, semirings, integral reductions.

1. Въведение

Универсалният агрегатор $A \circ$, дефиниран в [1], представлява общ математически оператор, който моделира еднотипен процес на обобщаване на множество данни чрез асоциативна бинарна операция. Класическите оператори сума Σ , произведение Π , минимум и максимум, логическите квантори \wedge, \vee , конкатенация на низове, интеграл в дискретен вид, семирингови редукции, Всички те могат да бъдат определени като специални реализации на общия оператор $A \circ$, когато се избере подходящ моноид $(S, \circ, e \circ)$, домейн D , филтър G и трансформация f .

Ключовата мотивация за този подход е, че унифицирането на толкова широк клас операции позволява стандартизирано, силно формално, машинно-съвместимо описание на изчисленията, независимо дали става дума за числови, логически, символни, матрични, аналитични или алгоритмични задачи. Това е особено важно в области като автоматизация на математически изчисления, формален анализ на алгоритми, оптимизационни схеми, интегрални, суматорни и пределни процедури, матрични и тензорни редукции, паралелни и разпределени изчисления, формални методи в компютърните науки и др.

За да бъде агрегаторът коректен и приложим във всички тези контексти, той трябва да удовлетворява определени структурни свойства, които гарантират стабилност, независимост от подредбата и възможност за разлагане на операции. Настоящата статия формализира тези свойства, обосновава тяхната необходимост и демонстрира приложението им чрез примери. [1], [2], [3], [5], [6], [7]

2. Да си припомним дефиницията на универсалния агрегатор $A \circ$

2.1. Домейн и множество на избор [1]

D — домейн, множество от всички възможни елементи x , върху които може да се итерираща.

$G : D \rightarrow \{0,1\}$ — предикат („гард“), който определя кои елементи участват в агрегацията.

Множеството на участващите елементи е

$$X = \{ x \in D \mid G(x) = 1 \}.$$

Предполага се, че X е крайно или ефективно изброимо.

2.2. Целева алгебрична структура [2], [3]:

Моноид – Тройка $(S, \circ, e \circ)$, състояща се от

Носител S — множество от стойности, които агрегаторът комбинира. Примери - $\mathbb{R}, \mathbb{R} \cup \{\pm\infty\}, \{0,1\}$, низове, матрици, множества, елементи в семиринг.

Асоциативна операция $\circ : S \times S \rightarrow S, (a \circ b) \circ c = a \circ (b \circ c)$.

Неутрален елемент $e \circ \in S, a \circ e \circ = e \circ \circ a = a$.

Ако допълнително $a \circ b = b \circ a$, моноидът е комутативен, което позволява агрегиране върху множества без ред.

2.3. Трансформация на стойности [1], [3]

$f: D \rightarrow S$ — функция, която преобразува елементите на домейна в стойности за комбиниране. Тя определя „какво измерваме“, „какво изчисляваме“ или „каква стойност извличаме“.

2.4. Формална дефиниция [1], [3]

Универсалният агрегатор е операторът $A_\circ[x \in D: G(x)] f(x)$

е дефиниран като:

$$A_\circ[x \in D: G(x)] f(x) = \begin{cases} e_\circ, & \text{ако } X = \emptyset, \\ f(x_1) \circ f(x_2) \circ \dots \circ f(x_n), & \text{ако } X = \{x_1, \dots, x_n\}. \end{cases}$$

При некомутативна операция \circ се приема фиксиран ред ρ върху X .

2.5. Концептуално тълкуване

Агрегаторът филтрира домейна чрез G , трансформира всеки валиден елемент чрез f , комбинира стойностите, чрез асоциативна операция \circ , като започва от неутралния елемент e_\circ и генерира една обобщена стойност за цялото множество.

3. Основни свойства на универсалния агрегатор A_\circ

Изследването на свойствата на агрегатора е необходимо, защото A_\circ е общ оператор, който трябва да бъде приложим в широк спектър от математически, алгоритмични и инженерни задачи. За да бъде той стабилен, предсказуем и годен за приложение в интегрални, оптимизация, паралелни алгоритми, обработка на матрици, семирингови редукции и други структури е критично да се изяснят и формализират тези свойства.

3.1. Асоциативност на операцията \circ [2], [3]

Условие - $(a \circ b) \circ c = a \circ (b \circ c) \forall a, b, c \in S$

Смисъл и тълкуване - Това означава, че групирането на стойностите при комбиниране няма значение. Агрегаторът може да изпълнява - ляво сгъване $((a \circ b) \circ c) \circ d$...; дясно сгъване $a \circ (b \circ (c \circ d) \dots)$; двоично/дървовидно сгъване и резултатът винаги е един и същ.

Пример: Сума: $(a + b) + c = a + (b + c)$. Произведение: също асоциативно. Матрици: $(AB)C = A(BC)$. Интегрални суми (дискретизация): $(\sum_i f(x_i)) + f(x_{i+1}) = \sum_i f(x_i) + f(x_{i+1})$

Асоциативността е фундаментална, поради няколко основни фактора:

- Позволява паралелни изчисления – можем да разделим домейна D на части, да изчислим частични агрегати и да ги комбинираме.
- Позволява оптимизации на порядък на изчисление – например при матрично умножение, редукции в GPU/CPU, интегрални апроксимации.
- Осигурява съвместимост с катаморфичен модел – fold операторът в програмните езици работи точно с асоциативни операции.
- Гарантира коректно поведение върху дървовидни структури – важен аспект при синтактични дървета, графи, автоматични доказателства.

3.2. Моноидност [2]

Това свойство комбинира асоциативност и наличие на неутрален елемент.

Условие $a \circ e = e \circ a = a$

Празната агрегация винаги връща „нулевата“ стойност за операцията

$$A \circ [\emptyset]f(x) = e$$

Пример за използване - Суми: $e = 0$; Произведения: $e = 1$; Минимум: $e = +\infty$; Максимум: $e = -\infty$; Конкатенация: празен низ;

Свойството позволява дефиниране на агрегатора върху произволни множества, включително празно множество. Гарантира стабилност в рекурсивни алгоритми, включително - динамично програмиране, divide-and-conquer, матрични редукции. Намира приложение при коректно дефиниране на пределни процедури, като - интеграл като граница на суми; серия като граница на частични суми

3.3. Комутативност (ако е налична) [2], [5], [6]

Условие $a \circ b = b \circ a$

Пример: Сума: $a + b = b + a$; Произведение: $ab = ba$; Мин/макс: $\min(a, b) = \min(b, a)$.

Комутативността не важи при матрично умножение, композиция на функции, конкатенация на низове, операции в несъразмерни семиринги.

Комутативност не се изисква, но ако я има, агрегаторът може да работи директно върху множества $A \circ [x \in D: G(x)]f(x)$, без да уточняваме реда.

Свойството дава независимост от реда на обхождане — критично при паралелни алгоритми; съвместимост с математическата интуиция (сумата на множество числа е еднозначна); коректно поведение в безкрайни редукции, където редът може да е неопределен; възможност за използване на агрегатора в модели за вероятности, тъй като там редът е произволен.

3.4. Идентичност и неутралност [2]

Това е друг начин да се формулира - $A \circ [x \in D: G(x)]e = e$

Това означава, че стойностите, равни на неутралния елемент, не влияят на резултата.

Използва се в интегрални дискретизации: добавянето на нулеви тегла не променя резултата. В матрични редукции: умножение по единична матрица не променя резултата. В логически конюнкции: $True \wedge a = a$.

Това свойство улеснява алгоритмичната реализация – може да добавяме dummy стойности без последствия. То позволява padding при паралелни блокови изчисления – напр. GPU kernels. Гарантира стабилност при стрийминг обработка – когато входът идва по части.

3.5. Инвариантност по отношение на предиката G [4], [5]

Ако: $\{x \in D: G_1(x) = 1\} = \{x \in D: G_2(x) = 1\}$

Тогава: $A \circ [x \in D: G_1(x)]f(x) = A \circ [x \in D: G_2(x)]f(x)$.

Смисълът е, че агрегаторът зависи само от множеството на валидните елементи, а не от това как е описано условието.

Пример: $G(x): x < 10$; $G'(x): x \leq 9$ - и двата дават едно и също множество.

Това свойство дава стабилност при промяна на формулировки на условията. То позволява оптимизации в компилатори и математически софтуер.

Позволява използване на различни логически описания на едно и също множество, което е важно в интегралите с различни параметризации, дискретни модели с различни индексации, матрични блокови структури и др.

3.6. Инвариантност по изоморфизъм на домейна [2, 4]

Ако $\phi: D_1 \rightarrow D_2$ е биекция и $G_2(\phi(x)) = G_1(x)$, тогава:

$$A \circ [y \in D_2: G_2(y)]f(y) = A \circ [x \in D_1: G_1(x)]f(\phi(x)).$$

Използва се например при преиндексиране на сума, промяна на параметризацията на интеграл, промяна на ред на матрични блокове и др.

Свойството позволява промяна на координатни системи без промяна на резултата. Агрегаторът остава „структурно стабилен“, независимо от нотацията. То е основно за символни манипулации и алгоритми, които работят с абстрактни структури.

3.7. Разделителност (декомпозиция) [3], [7]

За дизюнктни множества D_1 и D_2 :

$$A \circ [x \in D: G(x)]f(x) = A \circ [x \in D_1: G(x)]f(x) \circ A \circ [x \in D_2: G(x)]f(x)$$

Пример - разделяне на интеграл: $\int_a^c f(x) dx = \int_a^b f(x) dx + \int_b^c f(x) dx$, блоково матрично умножение, сумарна обработка на части от големи масиви и др.

Това свойство е критично за паралелни алгоритми и GPU. Т е основен принцип в MapReduce, Spark, паралелни редукции. Позволява разлагане на сложни интегралите и комбинирани на частични резултати.

3.8. Монотонност [6], [7]

Ако върху S има наредба \leq и \circ е монотонна, тогава:

$$f_1(x) \leq f_2(x) \forall x \Rightarrow A \circ [x \in D: G(x)]f_1(x) \leq A \circ [x \in D: G(x)]f_2(x)$$

Пример - Суми на положителни числа, интегриране на неотрицателни функции, матрични операции с PSD матрици, логическа импликация за квантори и др. Свойството е важно за доказателства на сходимост и граници. То е критично при оптимизация, числен анализ, машинно обучение (напр. свръх- или под-оценки). Позволява включване на агрегатора в монотонни итеративни схеми, напр. методи на Пикара, динамични програми.

Някои от тези свойства ще бъдат разгледани по-подробно в следващи публикации.

Изводи и заключения.

Универсалният агрегатор A° обединява широк клас операции чрез формално единна дефиниция, базирана на моноид. Това създава унифициран апарат за обработка на множество математически структури [1], [2], [3], [5], [7].

Свойствата асоциативност, моноидност, комутативност, идентичност, инвариантност и декомпозиция — гарантират стабилност на оператора при редукции върху множества, дискретни интегрални, матрични блокове и алгоритмични разлагания.

Агрегаторът A° е приложим далеч отвъд чисто числови изчисления, но и при работа с дискретен интеграл, мин/макс оператори, квантори, матрични блокови операции, тензорни редукции, семиринги, паралелни алгоритми (MapReduce, GPU редукции) и др.

Изследването на свойствата е фундаментално, защото те определят кога A° може да бъде правилно дефиниран върху множества, оптимизиран за паралелна обработка, еквивалентен на катаморфизъм от свободен моноид и използван в алгоритми със зададена коректност и монотонност

Статията може да се разглежда като основа за една обща теория на агрегиращите оператори, сравнима с теорията на интегрални, семиринги и катаморфизми.

References:

[1] Toshkov, A., *Дефиниране на универсален математически агрегатор*. КНК, т.14, 2025г. Бургаски свободен университет. (*Дефиниране на универсален математически агрегатор*)

[2] MAC LANE, Saunders., *Categories for the Working Mathematician*. Springer-Verlag, 1998.(Класически труд, дефиниращ моноиди, катаморфизми и универсални свойства.)

[3] BIRD, Richard; DE MOOR, Oege., *Algebra of Programming*. Prentice Hall, 1997.(Катаморфизми, редукции, fold оператори — фундаментално за агрегиране.)

[4] GOGUEN, Joseph; BURSTALL, Rod., *Institutions: Abstract Model Theory for Specification and Programming*. Journal of the ACM, 1984.(Универсални конструкции, абстрактни оператори.),

[5] GOLDBLATT, Robert., *Topoi – The Categorical Analysis of Logic*. Dover, 2006.(Формални оператори, квантори, структурни свойства.)

[6] KLEENE, Stephen., *Mathematical Logic*. Wiley, 1967.(Квантори, логически оператори – връзка с A° чрез специални моноиди.)

[7] HEBERT, Jean-Paul; SIMMONS, Harold., *Semirings and their Applications*. Springer, 2009.(Семиринги, редукции, дискретни интегрални — директно свързано.)