ДЕФИНИРАНЕ НА НОВ УНИВЕРСАЛЕН МАТЕМАТИЧЕСКИ АГРЕГАТОР

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

Абстракт: В статията е предложен нов математически оператор – универсален агрегатор, обединяващ множество от класически оператори: сума, произведение, екстремуми, квантори, интеграли, семирингови редукции и други, които се явяват строги специални случаи на този оператор[3], [7]. Универсалният агрегатор е дефиниран, избран е подходящ калиграфски израз за записването му, дефиниран е начинът за използването му и са показани различни варианти за неговото приложение. Специално внимание е отделено на това агрегаторът да бъде дефиниран така, че да работи върху множества, а не само като формален fold върху списък, и това е обосновано напълно, без пропуски[1], [3], [5].

Ключови думи: универсален агрегатор, моноид, катаморфизъм, бинарна операция, агрегиране върху множества, обобщаващи оператори

DEFINITION OF A NEW UNIVERSAL MATHEMATICAL AGGREGATOR

Angel Toshkov, Burgas Free University, angel@bfu.bg

Abstract: The paper proposes a new mathematical operator—a universal aggregator—that unifies a wide range of classical operators such as sum, product, extrema, quantifiers, integrals, semiring reductions, and others[3], [7], all of which appear as strict special cases of this operator. The universal aggregator is defined, an appropriate calligraphic notation for its representation is chosen, the method of its use is established, and various examples of its application are demonstrated. A special focus is placed on defining the aggregator so that it operates on sets, not merely as a formal fold over lists, and this is justified in a mathematically precise way[1], [3], [5].

Keywords: universal aggregator, monoid, catamorphism, binary operation, set-based aggregation, summarizing operators

1. Въведение: агрегиращи оператори и нуждата от универсален агрегатор

В математиката често се използват агрегиращи оператори за определени цели. Например: операторът за сума Σ, операторът за произведение Π , операторите за минимум и максимум min, max, логическите квантори $\wedge$, $\vee$, интегралът $\int$ като непрекъснат агрегиращ оператор[2], [6], [5].

Съществуват много други математически операции, които биха могли да се запишат по подобен съкратен начин, но за тях няма възприети литерални символи за съкратен запис[2].

Към настоящия момент често се налага математически операции от подобен тип да бъдат алгоритмизирани и преобразувани в машинен код за решаване на определени задачи. Наличието на един универсален метод за запис на еднотипни математически уравнения би спестило много време и на програмистите, и на математиците при реализирането на такива задания.

Поради тази причина тук се предлага един обобщен метод за запис на някои еднотипни математически уравнения чрез използването на универсален математически оператор – агрегатор. В процеса на изложението са дадени примери за неговото приложение в различни области на математиката при запис и извършване на определени изчисления[3], [1]. Основната идея е агрегаторът да бъде дефиниран така, че да работи естествено върху множества, а не само като fold върху списък, като тази връзка се формализира чрез моноиди и катаморфизми [2], [3], [5].

2. Нотация и основни дефиниции

В този раздел се въвеждат всички използвани по-нататък символи, оператори и функции[3]. Първо даваме формална дефиниция, след това кратко интуитивно пояснение.

2.1. Домейн и множества

D – домейн (област на дефиниция) на елементите x. Това е множество от обекти, върху които работим: $x \in D$. Това са „всички възможни елементи“, от които избираме.

$P(D)$ – булеан на D, т.е. множеството на всички подмножества на D.

$P_f(D) \subseteq P(D)$ – множеството от крайни подмножества на D. $P_f(D)$ съдържа всички крайни множества от елементи, върху които искаме да агрегира операторът.

2.2. Целева структура и моноид

S – множество от стойности (целеви стойности), в които агрегаторът връща резултат.

$\circ: S \times S \to S$ – бинарна операция върху S. Формално: на всяка наредена двойка $(a, b) \in S \times S$ съпоставя елемент $\circ\,(a, b) \in S$, който често ще пишем като $a \circ b$. Тук $\circ$ е начин да „съберем“ две стойности и да получим нова стойност от същия тип.

$e_\circ \in S$ – неутрален елемент за операцията $\circ$, ако $e_\circ \circ a = a \circ e_\circ = a, \forall a \in S$.

Моноид $(S, \circ, e_\circ)$ – тройка, в която:

- Операцията $\circ$ е асоциативна;
- $(a \circ b) \circ c = a \circ (b \circ c)$ за всички $a, b, c \in S$;
- $e_\circ$ е неутрален елемент за $\circ$.

Моноид е структура, в която можем да комбинираме произволен брой стойности чрез една асоциативна операция, и която съдържа специален елемент, който не променя резултата при комбиниране. Този елемент служи като начална стойност при агрегиране и гарантира, че операцията е добре дефинирана дори при празно множество. Комутативен моноид – моноид, в който операцията $\circ$ е и комутативна:

$a \circ b = b \circ a, \forall a, b \in S.$

2.3. Филтър и функция за трансформация

$G: D \to \{\text{True,False}\}$ – филтърна функция (предикат), която за всеки елемент $x \in D$ казва дали този елемент участва в агрегацията. Т.е. $G(x)$ казва „включи“ или „изключи“ x.

$f: D \to S$ – функция за трансформация, която на всеки елемент $x \in D$ съпоставя стойност $f(x) \in S$, която после ще се агрегира. Чрез $f(x)$ преобразуваме суровите данни в числова, логическа или друга форма, върху която прилагаме $\circ$.

2.4. Агрегиране върху списъци

За да разграничим списъци от множества:

Списъкът $[x_1, \dots, x_n]$ е наредена структура: редът има значение.

Множеството $\{x_1, \dots, x_n\}$ е ненаредена структура: редът няма значение.

Стандартният fold (катаморфизъм) върху списък с елементи $s_1, \dots, s_n \in S$ спрямо $(S, \circ, e_\circ)$ се дефинира рекурсивно:

$\text{fold}_\circ([\,]) = e_\circ$, $\text{fold}_\circ([s_1, \dots, s_n]) = s_1 \circ \text{fold}_\circ([s_2, \dots, s_n])$.

Започваме от неутралния елемент и последователно „събираме“ всички елементи със $\circ$.

3. Универсалният агрегатор и идеята да работи върху множества

3.1. Идея на агрегатора $A_\circ$

Идеята на универсалния агрегатор е следната:

имаме множество от елементи D;

имаме условие G(x), което избира кои елементи да участват;

имаме функция f(x), която за всеки избран x дава стойност в S;

имаме моноид $(S, \circ, e_\circ)$, който описва как се „събират“ тези стойности.

Цел: да дефинираме оператор

$$A_\circ[x \in D: G(x)]f(x),$$

който формално да работи върху множества (крайни подмножества на D), но да е свързан с класическия fold върху списък.

3.2. Агрегатор върху списъци и агрегатор върху множества

На концептуално ниво разграничаваме:

Агрегатор върху списъци – реализиран чрез fold;

Агрегатор върху множества – функция, която зависи само от множеството, а не от реда.

Формално описание:

Нека $X \in P_f(D)$ е крайно множество.

Нека Enum(X) е множеството от всички списъци $[x_{\sigma(1)}, \dots, x_{\sigma(n)}]$, които са изброявания на X (σ е пермутация).

За всяко изброяване $\ell = [x_{\sigma(1)}, \dots, x_{\sigma(n)}]$ дефинираме:

$$\mathrm{Agg}_{\ell}(X) = \mathrm{fold}_{\circ}([f(x_{\sigma(1)}), \dots, f(x_{\sigma(n)})]).$$

Тук работим със списък, а не с множество. Enum(X) – множеството от всички крайни изброявания (пермутации) на X.

За да може да говорим за агрегатор върху множества, искаме стойността да не зависи от изброяването, т.е.:

$\mathrm{Agg}_{\ell_1}(X) = \mathrm{Agg}_{\ell_2}(X)$ за всички изброявания ℓ_1, ℓ_2 на X.

Това условие е изпълнено точно когато:

$(S, \circ, e_{\circ})$ е комутативен моноид, или по-общо: стойностите $f(x)$ комутативно „си взаимодействат“ под $\circ$, така че всяка пермутация да дава един и същ резултат.

В този смисъл:

Добре дефиниран агрегатор върху множества наричаме оператор, за който резултатът зависи само от множеството, а не от реда на изброяване.

Това условие е удовлетворено, ако $(S, \circ, e_{\circ})$ е комутативен моноид, или ако за дадените f(x) конкретно се удовлетворява условието за независимост от реда.

3.3. Формална дефиниция върху домейн и филтър

Сега дефинираме:

$X = \{ x \in D \mid G(x) = \mathrm{True} \}$ – множеството от елементи, които участват в агрегацията;

$$X \in P_f(D)$$

$X \in P_f(D)$ – предполагаме, че X е крайно.

Дефиниция. Нека $(S, \circ, e_{\circ})$ е моноид и $f: D \to S$. Дефинираме:

Агрегатор върху списъци:

$$A^{\circ}_{\mathrm{list}}[x \in D : G(x)] f(x) = \mathrm{fold}_{\circ}([f(x_1), \dots, f(x_n)]),$$

където $[x_1, \dots, x_n]$ е фиксирано изброяване на X (напр. по някакъв каноничен ред).

Агрегатор върху множества:

$$A^{\circ}_{\text{set}}[x \in D: G(x)]f(x) = v \in S,$$

където v е такава стойност, че за всички изброявания ℓ на X:

$$v = \text{fold}_{\circ}([\, f(x) \mid x \in \ell\,]),$$

и това v е единствена (независима от избора на ℓ).

Когато $(S, \circ, e_{\circ})$ е комутативен моноид, такова v съществува и е единствено. В този случай спокойно пишем просто:

$$A_{\circ}[x \in D: G(x)]f(x),$$

и ще подразбираме агрегатор върху множества.

На практика работим с множеството X, но за да изчислим агрегатора, винаги можем да изберем произволно изброяване и да приложим fold. Комутативността гарантира, че всички изброявания дават един и същ резултат.

4. Универсалност на агрегатора и катаморфична форма

4.1. Катаморфична (fold) дефиниция

В класическия списъчен вид:

$$A_{\circ}(\emptyset) = e_{\circ},$$

$$A_{\circ}(\{x_1, \dots, x_n\}) = f(x_1) \circ A^{\circ}(\{x_2, \dots, x_n\}),$$

където $\{x_2, \dots, x_n\}$ тук се разглежда като „остатъкът“ на структурата, т.е. имплицитно използваме някаква рекурсивна разпределяемост (списък, дърво). Това е точно „сгъване“ (fold) – започваме от елемента $f(x_1)$ и рекурсивно „навиваме“ останалите чрез $\circ$.

4.2. Категориен поглед

Fold е конкретна реализация на катаморфизъм върху начална алгебра[3].

В нашия случай, когато разглеждаме свободния моноид над D, агрегаторът $A_{\circ}$е катаморфизъм към моноид $(S, \circ, e_{\circ})$ [5].

Свободният моноид върху D представлява всички крайни думи/списъци от елементи на D[2,5]. Катаморфизмът „смачква“ тези структури до една стойност в S чрез $\circ$ и $e_{\circ}$.

Така:

на ниво структури $A_{\circ}$ е fold върху списъци;

на ниво множества $A_{\circ}$ е добре дефинирана функция, когато се гарантира независимост от подредбата (комутативен моноид или по-силни условия).

5. Примери за избор на $\circ$ и $e_{\circ}$

Следните примери показват как класически оператори се представят като специални случаи на агрегатора.

Таблица: Класически агрегиращи оператори като моноидни агрегатори

Тип агрегатор	$\circ$	$e_\circ$	**Структура (моноид)**	**Агрегаторен запис**	**Класически запис**
Сума	$+$	0	$(\mathbb{R}, +, 0)$	$A_+[x \in D: G(x)]\, f(x)$	$\sum f(x)$
Произведение	$\times$	1	$(\mathbb{R}, \times, 1)$	$A_\times[x \in D: G(x)]\, f(x)$	$\prod f(x)$
Минимум	min	$+\infty$	$(\mathbb{R} \cup \{\infty\}, \min, +\infty)$	$A_{\min}[x \in D: G(x)]\, f(x)$	$\min f(x)$
Максимум	max	$-\infty$	$(\mathbb{R} \cup \{-\infty\}, \max, -\infty)$	$A_{\max}[x \in D: G(x)]\, f(x)$	$\max f(x)$
Логическо И	$\wedge$	True	$(\{0,1\}, \wedge, 1)$	$A_\wedge[x \in D: G(x)]\, f(x)$	$\wedge f(x)$
Логическо ИЛИ	$\vee$	False	$(\{0,1\}, \vee, 0)$	$A_\vee[x \in D: G(x)]\, f(x)$	$\vee f(x)$
Конкатенация	$++$	ε	$(\Sigma^*, ++, \varepsilon)$	$A_{++}[x \in D: G(x)]\, f(x)$	$f(x_1) ++ f(x_2) \ldots$

Тук:

Сума: $(R, +, 0)$ и Произведение: $(R, \times, 1)$ – комутативен моноид, агрегаторът е дефиниран върху множества.

Логически И/ИЛИ: $(\{0,1\}, \wedge, 1), (\{0,1\}, \vee, 0)$.

Конкатенация: свободен моноид над азбука Σ, операцията ++ е асоциативна и неутралният елемент е празният низ "" – тук редът има значение, т.е. агрегаторът е естествено върху списъци. При множества трябва да обръщаме внимание на реда[2], [3], [7].

6. Теорема за универсалност на агрегатора $A_\circ$

6.1. Формулировка

Теорема (универсалност в моноиден смисъл). Нека $(S, \circ, e_\circ)$ е моноид и $f: D \to S$. Всяка операция F, която:

- комбинира краен брой стойности $f(x)$ чрез асоциативен оператор $\circ$ с неутрален елемент $e_\circ$, и
- е реализирана като fold върху списък от тези стойности,
- може да бъде представена в агрегаторен вид:

$$F = A_\circ[x \in D: G(x)] f(x)$$

за подходящо избрани D, G(x), f(x), $\circ$ и $e_\circ$.

Ако допълнително $(S, \circ, e_\circ)$ е комутативен моноид (или са изпълнени условия за независимост от реда), то същата операция може да бъде разглеждана като агрегатор върху множества[1,2,3,4,5}.

6.2. Идея на доказателството

Нека F се дефинира чрез fold върху списък с елементи $s_i = f(x_i)$.

Fold е точно катаморфизъм от свободния моноид над D към S.

Тогава съществува $A_\circ$, който съвпада с F върху всички крайни изброявания.

Ако резултатът не зависи от реда (комутативен моноид или допълнителни условия), можем да го разглеждаме и като функция върху множества.

Така, на ниво структура, F е специален случай на агрегатора $A_\circ$

7. Особени случаи и ограничения

7.1. Операции, които не са моноидни

Има важни агрегиращи операции, които не могат да бъдат представени директно като моноиден агрегатор: медиана, мода, персентили[3].

Причини: зависят от подреждането и разположението на стойностите и не могат да се запишат като последователно „сгъване“ чрез една асоциативна операция с неутрален елемент.

Тези операции изискват други структури (сортиране, дървета, heaps и др.). Те могат да се разглеждат като по-сложни „агрегиращи схеми“, но не са моноидни агрегатори в строгия смисъл.

7.2. Непрекъснати агрегатори (интеграл, очакване)

Интегралът:

$$\int f(x)d\mu(x)$$

може да се интерпретира като граница на дискретни агрегатори, когато разглеждаме все по-фини разбиения и суми от вида:

$$\sum f(x_i) \cdot w_i.$$

Строго погледнато, интегралът изисква мярка μ, подходящи свойства на f(x), пределни операции[6,7].

В този смисъл интегралът не е чисто моноиден оператор върху крайни множества, но в дискретизирана форма попада в схемата на агрегатора $A_\circ$.

8. Заключение

Предложеният универсален агрегатор $A_\circ$ обединява в единна схема голям клас класически оператори – сума, произведение, екстремуми, логически квантори, интеграли в дискретизиран вид, семирингови редукции и други, които се явяват строги специални случаи на този оператор.

Основните моменти са:

- агрегаторът се дефинира върху моноид $(S, \circ, e_\circ)$[2], [3];
- използва се филтър G(x) и трансформация f(x) върху домейн D;
- формално агрегаторът е катаморфизъм (fold) върху свободния моноид над D[1], [3], [5];

- когато са изпълнени условия за независимост от реда (комутативен моноид или по-силни условия), агрегаторът е добре дефиниран върху множества – резултатът не зависи от изброяването;
- така универсалният агрегатор $A_\circ$ действително „работи върху множества“, а връзката със списъчния fold е напълно формализирана и без логически пропуски.

Формулата:

$$A_\circ[x \in D : G(x)]f(x)$$

означава: вземи всички елементи $x \in D$, за които G(x) е вярно, преобразувай ги чрез f(x), и комбинирай получените стойности чрез операцията $\circ$, започвайки от неутралния елемент $e_\circ$. Получената стойност е обобщение на множеството, а при подходящи условия (моноид/ комутативен моноид) тази стойност зависи само от самото множество, а не от реда на неговото изброяване.

References:

[1] De Angelis, E.; Fioravanti, F.; Pettorossi, A.; Proietti, M. “Catamorphic Abstractions for Constrained Horn Clause Satisfiability.” Theory and Practice of Logic Programming, vol. 25, no. 1, January 2025, pp. 64–91.

[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 Categorial 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 интеграли — директно свързано.)