

АРГУМЕНТАЦИЯ ПРИ СОСТАВЛЕНИИ АЛГОРИТМОВ

Цветков В.Я.

МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail:matcinvt@gmail.com

Цель работы – описание методики применения аргументации для обоснования работоспособности алгоритмов. Показано, что целесообразность применения аргументации появляется тогда, когда логическая схема не позволяет ответить на вопрос об обоснованности алгоритма. Показано, что схема аргументации является обоснованием алгоритма, но самим алгоритмом не является. Кратко описаны виды аргументации, применяемые для анализа алгоритмов. Дано сравнение логической и формальной аргументации. Основу методики, основанной на формальной аргументации, составляет аргументация Дунга. Даны три варианта аргументации для обоснования новых алгоритмов. Первый вариант возникает при расширении известного алгоритма в новый алгоритм за счет использования нового бесконфликтного подмножества алгоритма. Второй вариант имеет место, при сравнении нового и апробированного алгоритмов. Третий вариант аргументированного обоснования алгоритма имеет место, при одиночном включении операторов в известный алгоритм, при условии его бесконфликтности с остальной частью алгоритма. Описана целесообразность применения метода.

Ключевые слова: алгоритмы, вычисления, аргументация, логическая аргументация, формальная аргументация, аргументация Дунга, обоснованность алгоритма, конфликт, метки, логическая аргументация.

ARGUMENTATION IN DEVELOPING ALGORITHMS

Tsvetkov V. Ya.

MIREA - Russian Technological University, 119454, Moscow, 78 Vernadskogo Avenue, Russia, e-mail:matcinvt@gmail.com

The purpose of the work is to describe the methodology for applying argumentation to justify the performance of algorithms. It is shown that the expediency of using argumentation appears when the logical scheme does not allow answering the question about the validity of the algorithm. It is shown that the argumentation scheme is the substantiation of the algorithm, but is not the algorithm itself. The types of argumentation used for the analysis of algorithms are briefly described. A comparison of logical and formal argumentation is given. The basis of the methodology based on formal argumentation is Dung's argumentation. Three variants of argumentation are given to substantiate new algorithms. The first option arises when a known algorithm is extended into a new algorithm by using a new conflict-free subset of the algorithm. The second option takes place when comparing the new and proven algorithms. The third variant of the reasoned substantiation of the algorithm takes place, with a single inclusion of operators in a known algorithm, provided that it does not conflict with the rest of the algorithm. The expediency of applying the method is described.

Keywords: algorithms, calculations, argumentation, logical argumentation, formal argumentation, Dung's argumentation, algorithm validity, conflict, labels, logical argumentation.

Введение

Принято применять логику при составлении алгоритмов [1, 2]. Алгоритм при этом можно рассматривать как логическое следование или логическую цепочку [3]. Поэтому при составлении алгоритма сначала создают логическую схему, потом преобразуют ее в функциональную схему, которая осуществляет вычисление. Если логика корректно описывает вычислительный процесс, то такая программа работает корректно. Но если логическая схема упрощенно описывает реальные вычисления, то в процессе работы появляются свои вычисления, которые можно обозначить термином конфликт. Такую программу начинают отлаживать, внося изменения в алгоритм и программное обеспечение. Первоначальная логика уже не может служить основой корректности алгоритма. Для проверки корректности отлаженной программы необходимо применять теорию аргументации [4, 5]. При правильном алгоритме наборы данных также влияют на результат вычислений. Если в процессе вычислений появляется машинный ноль, то результаты вычислений становятся некорректными. Появление машинного нуля можно рассматривать как конфликт. При обработке данных с погрешностями, при определенной величине погрешностей, может появляться несопоставимость результатов вычислений. Эту ситуацию также следует считать конфликтной. В практике вычислений может появиться ситуация, когда

решенную задачу с известным алгоритмом требуется усложнить и проще модифицировать известный алгоритм, чем писать всю программу заново. В этом случае говорят о расширенном множестве операторов, имея в виду модифицированный алгоритм и программу. В практике вычислений может появиться ситуация, когда создается новый алгоритм, структурно схожий с апробированным и надежным алгоритмом, решающим другую задачу. В этом случае представляет интерес оценка надежности нового алгоритма по известному алгоритму. Аргументация также может быть представлена аргументационное следование. Аргументационное следование является более общим понятием чем логическая цепочка. Аргументация может быть логической, формальной, коммуникационной, когнитивной, когнитивно-коммуникационной. Логическая цепочка есть частный случай аргументационной цепочки. В тех случаях, когда чистая логика не работает доказательством надежности и условной корректности алгоритма становится аргументация. При составлении алгоритмов допустима не любая аргументация, а только логическая, формальная и коммуникационная. Эти виды аргументаций служат основой алгоритмической аргументации. Коммуникационная аргументация применяется при эвристической отладке. При наличии конфликтов применяют формальную аргументацию

Алгоритмическая аргументация

Алгоритм можно рассматривать как логическую цепочку или в более общем виде как аргументационную цепочку [6] или аргументационное множество. Алгоритм можно рассматривать упорядоченную совокупность информационных аргументационных единиц. Новое понятие аргументационная информационная единица описывает единицу аргументации. Упорядоченное множество информационных аргументационных единиц называют аргументационной цепочкой. Ее аналогом является логическая цепочка. Различие между ними в том, что логическая цепочка показывает логическую связь, аргументационная цепочка доказывает обоснованность связанных аргументов. Логическая цепочка использует логические единицы [7], аргументационная цепочка использует аргументационные единицы [8]. Аргументационная цепочка может быть конфликтной и бесконфликтной. Бесконфликтная аргументационная цепочка соответствует и описывает детерминированные алгоритмы. Она соответствует секвенциям. Конфликтная аргументационная цепочка описывает недетерминированные алгоритмы или алгоритмы, которые работают со сбоями.

Среди различных видов аргументации можно выделить ту, которая связана с алгоритмами и назвать ее алгоритмическая аргументация. Алгоритмическая аргументация может быть рассмотрено как подмножество теории аргументации применительно составлению и анализу алгоритмов. Алгоритмическая аргументация использует следующие понятия: аргументационные информационные единицы, аргументационное множество переменных, отношения, алгоритмические схемы, информационную алгоритмическую ситуацию [9, 10] и алгоритмический морфизм. В алгоритмическую аргументацию входят логическая (если ее можно применить), формальная (специальная аналитика) и коммуникационная (эвристические и статистические методы при отладке).

Информационная единица алгоритма обычно описывает процесс. Топологически она соответствует ориентированной дуге. Аргументационная информационная единица содержит две переменные и отношение между ними. Топологически она соответствует графу с двумя вершинами и ориентированной дугой между ними. В силу этого аргументационные информационные единицы называют также аргументационными парами. Аргументационные информационные единицы описывают наличие или отсутствие конфликта. Они могут описывать его уменьшение или увеличение. Отсутствие конфликта соответствует логической единице.

Аргументационное множество A может быть рассмотрено как описание схемы алгоритма или как информационная ситуация вокруг алгоритма. Аргументационное множество A называется бесконфликтным, если входящие в него аргументационные пары являются комплементарными. Другими словами, отношением конфликта между парами на данном множестве является пустым. Аргументационное множество A называется конфликтным, если входящие в него аргументационные пары не являются комплементарными. Другими словами, отношением конфликта между некоторыми парами на данном множестве не является пустым. Конфликт возникает при наличии минимум двух противодействующих пар.

Формальный аргументационный подход с использованием метода аргументации Дунга.

На практике для аргументирования широко применяют аргументацию Дунга [11, 12]. В ней бинарное отношение в A , вызывающее конфликт (сбой) называется атака и обозначается как элемент множества R . На рис.1. приведен пример конфликтного аргументационного множества.

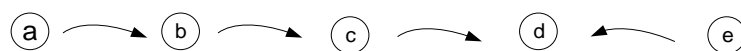


Рис.1. Конфликтное аргументационное множество

На рис.1 показана система аргументации $S = \langle A, R \rangle$, включающая аргументационное множество $A = \{a, b, c, d, e\}$ и совокупность отношений атаки $R = \{(a, b), (b, c), (c, d), (e, d)\}$. Она содержит пять аргументов (a, b, c, d, e) и четыре атаки. Атаки изображаются дугами в отличие от прямой стрелки, обозначающей импликацию. Аргумент, а атакует аргумент b, аргумент b атакует аргумент c, аргумент c атакует аргумент d, аргумент «e» атакует аргумент d. Применительно к алгоритмической схеме d может обозначать алгоритм или его часть. Аргументы a, b, c, e могут снижать надежность d алгоритма.

Дунг так определяет некоторые положения аргументации. Он вводит понятие аргументационное множество A и аргументационное подмножество или набор аргументов E. Он рассматривает произвольный аргумент a и определяет следующее:

- Аргумент $a \in A$ приемлем в отношении подмножества $E \subseteq A$ если и только если E защищает a. Это означает, что есть конфликтный аргумент b такой, что $\forall b \in A$ есть отношение $(b, a) \in R$.

- Дополнительно существует аргумент c, такой, что $\exists c \in E (c, b) \in R$.

- Набор аргументов E является бесконфликтным, если между его аргументами нет атаки, это означает, что $\forall a, c \in E, (a, c) \notin R$.

- Набор аргументов E допустим или E есть допустимое множество, если и только если он бесконфликтен и все его аргументы приемлемы в отношении E.

Чтобы решить, может ли аргумент (фактор алгоритма) быть принят или нет) в программу) или несколько аргументов могут быть приняты вместе (в программу), Dung определяет несколько семантик принятия, которых позволяет, с учетом системы аргументации, вычислять наборы аргументов (называемые *расширениями*). Например, для системы $S = \langle A, R \rangle$:

- Расширение H является полным продолжением S только если это допустимое множество и каждый приемлемый аргумент (a) относительно E принадлежит H.

- Расширение H является предпочтительным продолжением S только если это максимальный элемент $H \subseteq S$ среди допустимых множеств по S,

- H является стабильным продолжением S только если это бесконфликтное множество, которое атакует каждый аргумент (a), который не принадлежит H (формально, $\forall a \in A \setminus H, \exists b \in H$ такой, что $(b, a) \in R$,

- H является (уникальным) обоснованным продолжением S только если это наименьший элемент (относительно включения в набор) среди полных расширений S,

- Расширение H расширяет бесконфликтное множество E и фактически говорит о том, что алгоритм с таким расширением является аргументированным и обоснованным.

Еще одним подходом к аргументированию алгоритма является применение меток. Метки являются дополнительным способом, чтобы выразить принятие аргументов. Маркировка как совокупность меток - это отображение, которое связывает каждый аргумент с меткой *in* (аргумент принят), *out* (аргумент отклонен) или *undec* (аргумент не определен - не принят и не отклонен). Можно также отметить маркировку как набор пар аргумент, метка. Совокупность маркировки описания расширения множества называют картированием. Картирование - это восстановление маркировки в системе если и только если :

- $\forall a \in A, L(a) = in$ если и только если $\forall b \in A$ такой, что $(b, a) \in R, L(b) = out$

- $\forall a \in A, L(a) = out$ если и только если $\exists b \in in A$ такой, что $(b, a) \in R, L(b) = in$

- $\forall a \in A, L(a) = undec$ если и только если $L(a) \neq in$ и $L(a) \neq out$

Маркировку выполняет отладчик программы. Можно преобразовать каждое расширение в маркировку: аргументы расширения H находятся в *in*, а остальные находятся в *out*. Можно построить расширение H с помощью маркировки просто дополняя множество аргументами в *in*. Каминада [13, 14] доказал, что метки описания и полные расширения могут отображаться биективным образом. Более того, семантика может быть связана с некоторыми конкретными наборами меток описания.

Маркировка описания отличает аргументы, не принятые, потому что они атакованы принятыми аргументами от неопределенных аргументов, то есть от тех, которые не защищены, не могут защитить себя. Аргумент является *undec*, если он атакован как минимум другим *undec*. Если он атакован только аргументом *out*, то он должен быть *in*. Если он атакован каким-то аргументом *in*, то он есть *out*. Эти методы используются для аргументации алгоритмического расширений. Уникальная маркировка описания, которая соответствует системе S (рис.1) выше есть $L = \{(a, in), (b, out), (c, undec), (d, out), (e, in)\}$.

Можно и дальше продолжать теоретические выкладки, но остановимся на простом применении данной теории. Оно заключается в том, что если существуют две близкие программы или два схожих алгоритма и один, уже проверен на практике, то он может служить для аргументирования надежности другого алгоритма. Это вытекает из правил эквивалентности между структурами аргументации.

Существует несколько критериев эквивалентности между структурами аргументации. Большинство из этих критериев касаются наборов расширений или набора принятых аргументов. Назовем систему аргументации расширенной если $S1=S+H1$ $S2=S+H2$. Формально, учитывая семантику по Дунгу:

- две структуры аргументации эквивалентны, если они имеют одинаковый набор расширения, то есть $S1 \equiv S2 \Leftrightarrow H1 = H2$;

- две структуры аргументации эквивалентны, если они принимают скептически [15] одинаковые аргументы, то есть $S1 \equiv S2 \Leftrightarrow Sc\sigma(S1) = Sc\sigma(S2)$;

- две структуры аргументации эквивалентны, если они доверчиво [16] принимают одни и те же аргументы, то есть $S1 \equiv S2 \Leftrightarrow Cr\sigma(S1) = Cr\sigma(S2)$;

В заключении отметим основы логической аргументации. Логическая аргументация связана с понятием выводимость. Логический аргумент - это A пара, где первая левая часть представляет собой минимальный непротиворечивый набор формул, достаточный для вывода формулы для второй части аргумента. В работе [17] отмечается, что семантическое следование равносильно выводимости. Это подчеркивается сходством символов. Существует символ \vdash выводимость и аналогичный \vDash символ логического следования. В соответствии с законами логики логическое следование исключает неопределенность, модальность и вопросительные предложения. Семантическое понятие логического следования введено Тарским в 1936 г., и отражено в ряде поздних публикаций [18]. Формальное определение логического следования имеет вид.

$$(A_1, \dots, A_n) \vDash B \quad (1)$$

Семантическое определение логического следования (1) звучит так :Из аргументов (A_1, \dots, A_n) логически следует высказывание B , если не может быть так, что аргументы A_1, \dots, A_n истинны, а высказывание B – ложно, (т.е. если B истинно в любой модели, в которой истинны A_1, \dots, A_n . Отличительной чертой логического следования является то, что оно ведет от истинных высказываний только к истинным.

Заключение

Целесообразность применения аргументации появляется тогда, когда логическая схема не позволяет ответить на вопрос об обоснованности алгоритма. Система и схема аргументации является обоснованием алгоритма, но самим алгоритмом не является. Она ближе к понятию информационная конструкция, в то время как, алгоритм ближе к понятию прикладная модель. Схема аргументации в виде цепочки соответствует концептуальной или аргументационной схеме алгоритма, но она предназначена не для решения вычислительных задач, а для доказательства корректности и обоснованность этапов алгоритмического вычисления. При использовании аргументации необходимо выявлять события, которые можно назвать конфликтами и использовать их при анализе аргументов. Если схема аргументации алгоритма является бесконфликтной, то это говорит об обоснованности и реализуемости алгоритма. Если схема аргументации алгоритма является конфликтной, то это говорит о нецелесообразности применения данного алгоритма. В данной работе дано обоснование схемы алгоритма при его расширении. В данной работе дано обоснование схемы алгоритма при сравнении его с обоснованным алгоритмом. В данной работе дано описание механизма меток, которые позволяют отвечать на вопрос включать данный оператор в программу или не включать. В отличие от логических схем аргументационная схема допускает понятие конфликт, но эти конфликты анализирует и помогает устранять. Данная работа основан на формальном подходе с использованием аргументации Дунга. За рамками статьи остался подход Стивена Тулмина [19], заключающийся в применении для аргументации модели полей аргументов. За рамками статьи остался коммуникационный аргументационный подход. Это материал для дальнейших исследований.

Список литературы

1. Кудж С.А., Цветков В.Я. Логика и алгоритмы. – М.: МАКС Пресс, 2019. – 112 с.
2. Цветков В.Я. Алгоритмическая логика // Славянский форум. 2021, 4(34). С. 142-150.
3. Perdicoulis A., Batista L., Pinho P. Logical chains in territorial impact assessment //Environmental Impact Assessment Review. – 2016. – Т. 57. – С. 46-52.
4. Van Eemeren F. H., Grootendorst R., Kruijer T. Handbook of argumentation theory: A critical survey of classical backgrounds and modern studies. – Walter de Gruyter GmbH & Co KG, 2019.
5. Walton D. Argumentation theory: A very short introduction //Argumentation in artificial intelligence. – Springer, Boston, MA, 2009. – С. 1-22.
6. Rach N., Minker W., Ultes S. Increasing the naturalness of an argumentative dialogue system through argument chains //Computational Models of Argument. – IOS Press, 2020. – С. 331-338
7. Ожерельева Т.А. Логические информационные единицы // Славянский форум, 2015. - 2(8). - с.240-249.
8. Alvarado S., Dyer M. G., Flowers M. Editorial comprehension in oped through argument units. – UCLA

Computer Science Department, 1986. – С. 250-256.

9. Титов Е. К. Ситуационные вычисления // Славянский форум. -2020. – 4(30). -с.134-143.
10. Цветков В.Я. Систематика информационных ситуаций // Перспективы науки и образования. - 2016. - №5 (23). - с.64-68.
11. Dung P. M. An argumentation-theoretic foundation for logic programming //The Journal of logic programming. – 1995. – Т. 22. – №. 2. – С.151-177.
12. Dung P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games //Artificial intelligence. – 1995. – Т. 77. – №. 2. – С. 321-357.
13. Caminada M., Sakama C. On the existence of answer sets in normal extended logic programs //ECAI 2006, PROCEEDINGS. – 2006. – Т. 141. – С. 743-744.
14. Caminada M. Well-founded semantics for semi-normal extended logic programs //Proc. 11th Int'l Workshop on Nonmonotonic Reasoning. – 2006. – С. 103-108
15. Dung P. M., Mancarella P., Toni F. Computing ideal sceptical argumentation //Artificial Intelligence. – 2007. – Т. 171. – №. 10-15. – С. 642-674.
16. D. Gartner, F. Toni, CaSAPI: a system for credulous and sceptical argumentation, in: G. Simari, P. Torroni (Eds.), Proc. Workshop on Argumentation for Non-Monotonic Reasoning, 2007, pp. 80–95.
17. Верещагин Н.К., Шень А. Лекции по математической логике и теории алгоритмов. Часть 2. Языки и исчисления. — 4-е изд., испр. — М.: МЦНМО, 2012. — 240 с.
18. A. Tarski. On the concept of logical consequence // Logic, Semantics, Metamatematics, second edition, pp. 409-420. Indianapolis: Hackett, 1983.

References

1. Kudzh S.A., Tsvetkov V.YA. Logika i algoritmy. – М.: MAKS Press, 2019. – 112 s
2. Tsvetkov V.YA. Algoritmicheskaya logika // Slavyanskij forum. 2021, 4(34). S. 142-150.
3. Perdicoulis A., Batista L., Pinho P. Logical chains in territorial impact assessment //Environmental Impact Assessment Review. – 2016. – Т. 57. – С. 46-52.
4. Van Eemeren F. H., Grootendorst R., Kruiger T. Handbook of argumentation theory: A critical survey of classical backgrounds and modern studies. – Walter de Gruyter GmbH & Co KG, 2019.
5. Walton D. Argumentation theory: A very short introduction //Argumentation in artificial intelligence. – Springer, Boston, MA, 2009. – С. 1-22.
6. Rach N., Minker W., Ultes S. Increasing the naturalness of an argumentative dialogue system through argument chains //Computational Models of Argument. – IOS Press, 2020. – С. 331-338
7. Ozherel'eva T.A. Logicheskie informacionnye edinicy // Slavyanskij forum, 2015. - 2(8). - s.240-249.
8. Alvarado S., Dyer M. G., Flowers M. Editorial comprehension in oped through argument units. – UCLA Computer Science Department, 1986. – С. 250-256.
9. Titov E. K. Situacionnye vychisleniya // Slavyanskij forum. -2020. – 4(30). -s.134-143.
10. Tsvetkov V.YA. Sistematika informacionnyh situacij // Perspektivy nauki i obrazovaniya. - 2016. - №5 (23). - s.64-68.
11. Dung P. M. An argumentation-theoretic foundation for logic programming //The Journal of logic programming. – 1995. – Т. 22. – №. 2. – С.151-177.
12. Dung P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games //Artificial intelligence. – 1995. – Т. 77. – №. 2. – С. 321-357.
13. Caminada M., Sakama C. On the existence of answer sets in normal extended logic programs //ECAI 2006, PROCEEDINGS. – 2006. – Т. 141. – С. 743-744.
14. Caminada M. Well-founded semantics for semi-normal extended logic programs //Proc. 11th Int'l Workshop on Nonmonotonic Reasoning. – 2006. – С. 103-108
15. Dung P. M., Mancarella P., Toni F. Computing ideal sceptical argumentation //Artificial Intelligence. – 2007. – Т. 171. – №. 10-15. – С. 642-674.
16. D. Gartner, F. Toni, CaSAPI: a system for credulous and sceptical argumentation, in: G. Simari, P. Torroni (Eds.), Proc. Workshop on Argumentation for Non-Monotonic Reasoning, 2007, pp. 80–95.
17. Верещагин Н.К., Шень А. Лекции по математической логике и теории алгоритмов. Часть 2. Языки и исчисления. — 4-е изд., испр. — М.: МЦНМО, 2012. — 240 с.
18. A. Tarski. On the concept of logical consequence // Logic, Semantics, Metamatematics, second edition, pp. 409-420. Indianapolis: Hackett, 1983.