

ОБЛАСТИ ПРИМЕНИМОСТИ НЕСЕКВЕНЦИАЛЬНЫХ АЛГОРИТМОВ

Козлов А.В., Титов Е.К.

«МИРЭА - Российский технологический университет», 119454, Россия, г. Москва, проспект Вернадского, 78

Цель работы – исследование области применимости и неприменимости несеквенциальных алгоритмов. Раскрывается содержание понятия секвенция. Показана его полисемия и применение в программировании. Показано различие между секвенциальными и несеквенциальными программами. Приводится аналогия с детерминированными и не детерминированными вычислениями. Описаны принципы работы секвенциальных программ. В качестве несеквенциальных алгоритмов рассмотрены ситуационные и субсидиарные алгоритмы. На примере конструкции IF-THEN-ELSE показано различие в ее реализации в секвенциальных, в ситуационных и субсидиарных программах. Статья показывает, что применение несеквенциальных алгоритмов и методов позволяет решать большее число задач, которые секвенциальные программы не решают. Статья дает критерии применимости ситуационных и субсидиарных алгоритмов

Ключевые слова: алгоритмы, вычисления, секвенция, логическая последовательность, секвенциальные вычисления, ситуационные вычисления, субсидиарные вычисления, информационная конструкция, условная конструкция.

APPLICABILITY AREAS OF NON-SEQUENTIAL ALGORITHMS

Kozlov A.V., Titov E.K.

«MIREA - Russian Technological University», 119454, Moscow, 78 Vernadskogo Avenue, Russia

The purpose of the work is to study the area of applicability and inapplicability of non-sequential algorithms. The content of the concept of sequence is revealed. Its polysemy and application in programming are shown. The difference between sequential and non-sequential programs is shown. An analogy with deterministic and non-deterministic calculations is given. The principles of operation of sequential programs are described. Situational and subsidiary algorithms are considered as non-sequential algorithms. The example of the IF-THEN-ELSE construct shows the difference in its implementation in sequential, situational and subsidiary programs. The article shows that the use of non-sequential algorithms and methods allows solving a greater number of problems that sequential programs do not solve. The article gives the criteria for applicability of situational and subsidiary algorithms

Keywords: algorithms, computations, sequence, logical sequence, sequential computations, situational computations, subsidiary computations, information construction, conditional construction.

Введение

Секвенция (sequence) -полисемическое понятие, которое применяется в музыке, архитектуре, логике и программировании [1-3]. Секвенции широко применяют в биоинформатике [4, 5]. Синонимом секвенции в программировании является последовательность. В логике логическое следование или логическая цепочка [6]. В программировании [7, 8] секвенцией называют последовательность действий, необходимую для решения задачи. По существу, это определение близко к понятию линейного алгоритма или алгоритма первого рода. Однако между секвенцией и алгоритмом есть различие. «Алгоритм» трактуют определить как упорядоченную последовательность процессуальных шагов для решения конкретной задачи [9]. Порядок в последовательности шагов важен. Секвенция в программировании более широкое понятие. Оно относится к упорядоченному набору инструкций или задач. Инструкция может соответствовать процессуальному шагу, но может быть его описанием. Секвенция задач еще более общее описание по сравнению с процессуальными шагами. Существует понятие секвенциальное программирование [10, 11]. Секвенциальное программирование можно определить как логически корректное программирование с использованием логических следований и функциональных последовательностей. Секвенция близка детерминированным алгоритмам. Детерминированный алгоритм или секвенциальный алгоритм при одинаковых входных данных всегда дает один и тот же результат. В отличие от него недетерминированный алгоритм для одних и тех же входных данных может давать разные результаты. Это обусловлено средой, воздействующей на вычисления. Для секвенциальных алгоритмов такое влияние исключают и считают его недопустимым. Понятие недетерминированный алгоритм введено Р. У. Флойдом в

1967 году. [12]. Недетерминированные алгоритмы есть несеквениальные алгоритмы. Недетерминированные алгоритмы бывают разных типов. Рассмотрим два типа несеквениальных алгоритмов ситуационные и субсидиарные.

Секвениальные и несеквениальные методы

Секвениальное программирование точно отражает логическое мышление. В этом его преимущество и возможность оперативного анализа. Если вычисление нельзя логически сконструировать, приходится применять несеквениальные алгоритмы.

Можно рассмотреть два типа несеквениальных алгоритмов: ситуационные [13, 14] и субсидиарные [15, 16]. Ситуационный алгоритм допускает изменение условия при сохранении цели. Например, вычисление площади фигур разной формы, в том числе неправильной формы, может осуществляться геометрически, путем интегрирования или статистическими методами [17]. Субсидиарные алгоритмы применяют в живых системах, например деление живых систем является субсидиарной процедурой и выполняется по необходимости. Алгоритм принятия решений войсковым подразделением, для которого возникли непредвиденные условия, является несеквениальным, а субсидиарным.

Многие управляющие структуры алгоритмов используют условную информационную конструкцию (IF-THEN, -ELSE). Это информационная конструкция, которая по-разному реализуется в секвениальных и несеквениальных программах. Однако она по-разному применяется в секвениальных и несеквениальных программах. В секвениальных программах эта конструкция выполняется на уровне соседних шагов или процессов. она представляет собой развилку рис.1.

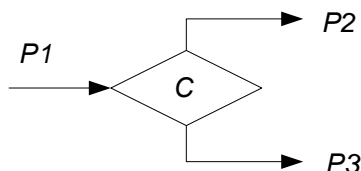


Рис.1. Условная конструкция в секвениальной программе

Секвениальная программа выполняется по принципу Step By Step [18]. Поэтому для нее рис.1 интерпретируется так

Step P1, If C, Then Step P2,-Else Step P3, (1)

Выражение (1) является фрагментом конкретной программы и полной интерпретацией условной конструкции. Важно подчеркнуть, что условная конструкция в секвениальной программе работает на уровне соседних процессов. принцип Step By Step формирует цепочку последовательных действий (рис.2).

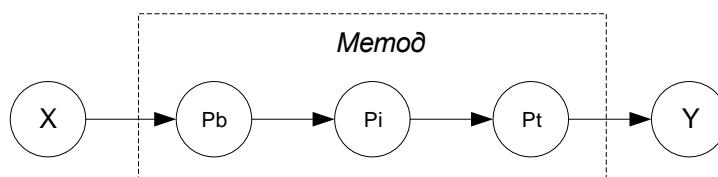


Рис.2. Последовательность выполнения секвениальной программы

На рис.2 показано что входные данные X подвергаются обработке в последовательности процессов от начального Pb до конечного Pe, что приводит к получению результата (выходных данных) Y. Упрощенно рис.2 описывается парадигмой «вход – метод – результат». В этой простой парадигме не говорится об условиях, их совмещают с данными или считают неизменными. В этой парадигме не говорится о том, что результат имеет качественное и количественное значение, например объем и площадь представляют собой разные категории и разное качество.

Управляющая структура ситуационного алгоритма также использует информационную конструкцию IF-THEN-ELSE позволяет ситуационному алгоритму выбирать альтернативу одну из многих. Однако эта конструкция применяется на уровне условия и результата. Ситуационный алгоритм является более чувствительным по сравнению с секвениальным алгоритмом (рис.2). Он допускает возможность изменения

условия и учитывает качество результата. На рис.3 показана реализаций условной конструкции IF-условия THEN- метод ELSE другой метод в ситуационной программе.

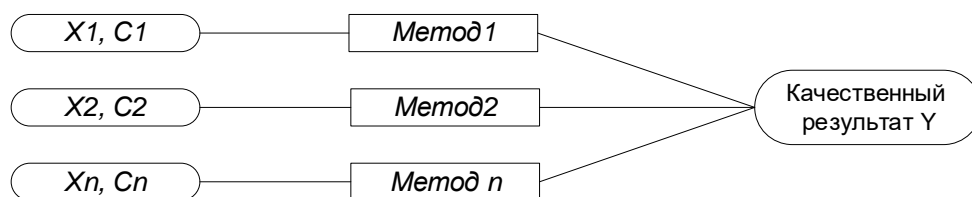


Рис.3. Условная конструкция в ситуационной программе

Поэтому ситуационного алгоритма конструкция имеет другой вид

Step P0, IF (Xi, Ci), THEN (метод i), ELSE switch (2)

Выражение (2) интерпретируется так. На начальном шаге анализируются условия и данные. В зависимости от них выбирается метод решения задачи иначе на переключатель.

Например, необходимо определить площадь фигуры. Фигурой может быть: треугольник, ромб, квадрат, круг, функция и ось X, неправильная фигура. Для расчета площади выбирают свой метод расчета, связанный с типом фигуры. Качественный результат будет одинаковым. Будет определена площадь фигуры. Но количественно значение может быть разным. Это не всегда важно. Круг может иметь большую площадь чем треугольник, дав прямоугольника могут иметь разную площадь. Но задача вычисления площади решена. При ситуационном вычислении связаны множество решений одной задачи. Упрощенно рис.3 описывается парадигмой «связанное семейство входов – связанное семейство методов – результат». Область использования ситуационного алгоритма намного шире, чем секвенциального алгоритма.

Еще больше условная конструкция развивается в субсидиарных алгоритмах. Для субсидиарных алгоритмов информационная конструкция IF-THEN-ELSE трансформируется в сложное выражение. Для субсидиарных алгоритмов условная конструкция реализуется так

Step P0 [SG(Lg), Sec(P)] → Step Pi, IF (Pi +Ei ≠Lgi), THEN (Pk≠Pi+1),
ELSE (SG CHANGE NG) (3)

Выражение (3) интерпретируется так. На начальном шаге P0 существует глобальная цель, которая включает множество локальных целей Lg для каждого этапа вычислений. Существует секвенция методов Sec(P) для достижения цели. На каком-то (не на начальном) этапе возникает ситуация вычислений, для которой процесс этапа Pi плюс воздействие внешней среды Ei исключают достижение локальной цели этого этапа Lgi. В этом случае либо осуществляют переход к качественно иному методу Pk, который не соответствует логическому следованию процесса Pi+1 и не входит в секвенцию методов Sec(P), либо происходит изменение глобальной цели SG на новую цель NG. Примером может служить дождевой червь. Он как система растет и развивается, но при определенных условиях делится на две системы.

Субсидиарные вычисления являются более общими по отношению к ситуационным вычислениям. В ситуационных вычислениях анализ ситуации является первым этапом. В субсидиарных вычислениях анализ ситуации может возникнуть на любом этапе. В ситуационных вычислениях цель всегда одна, и она достигается выбором методов в зависимости от ситуации. В субсидиарных вычислениях существует условная стратегическая цель. Но она может быть изменена в зависимости от обстоятельств.

В ситуационных вычислениях существует набор секвенций методов, который применяют при решении задачи. В субсидиарных вычислениях возможен переход от одной секвенции к другой.

В секвенциальных вычислениях используют одну секвенцию методов, которая служит основой достижения цели или решения задачи. Но это достигается путем наложения ограничений на изменение начальных условий или изменения условий в процессе вычислений. В случае таких изменений секвенциальные вычисления неприменимы. Но в этом случае применимы несеквенциальные вычисления. следовательно несеквенциальные вычисления расширяют условия и возможности решения задач и возможности управления с помощью таких алгоритмов.

Заключение

Область применимости несеквенциальных алгоритмов более широкая по сравнению с областью применения секвенциальных алгоритмов. Но целесообразность их применения ограничивается условиями вычислений. Точно также как параллельные вычисления не целесообразно применять для простых вычислений, также и несеквенциальные алгоритмы нецелесообразно применять для прямых вычислений. Область их применения определяется вариантами конструкции IF-THEN-ELSE. Если эта конструкция классическая и процессуальная, то целесообразней применять прямые алгоритмы. Если конструкция IF-THEN-ELSE процессуальная и имеет вид Step P0, IF (Xi, Ci), THEN (метод i), ELSE switch, то секвенциальные алгоритмы не применимы, а применимы ситуационные алгоритмы. Если конструкция IF-THEN-ELSE процессуальная и имеет вид Step P0 [SG(Lg), Sec(P)] → Step Pi, IF (Pi +Ei ≠Lgi), THEN (Pk≠Pi+1), ELSE (SG CHANGE NG), то секвенциальные алгоритмы не применимы, а применимы субсидиарные алгоритмы. Термин секвенциальные вычисления более мягкий и общий по отношению к термину детерминированные вычисления. Детерминированность подразумевает причинно-следственную связь. Секвенция предполагает наличие любого следования. например в Марковской цепи. Несеквенциальные вычисления позволяют решать более широкий круг вычислительных задач и задач управления. Общим для секвенциальных и несеквенциальных вычислений является то, что вычисления делятся на этапы и каждому этапу для его преодоления соответствует процесс. Но механизм решения задач существенно различается. Это подчеркивается на информационной конструкции условного перехода IF-THEN-ELSE. Эта информационная конструкция реализуется в разные схемы. Для секвенциальных вычислений она реализуется на определенном этапе вычислений в схему IF условие THEN процесс 1 ELSE процесс 2. Для ситуационных (несеквенциальных) вычислений она реализуется на первом этапе вычислений в схему IF-условия THEN- метод ELSE другой метод. Для субсидиарных (несеквенциальных) вычислений она реализуется на любом этапе вычислений в схему IF-невозможность достижения локальной цели THEN- либо метод другой секвенции ELSE изменение стратегической цели. Несеквенциальные вычисления требуют больших временных затрат, но решают задачи, которые секвенциальными методами решить нельзя. Это направление требует дальнейших исследований.

Список литературы

1. Sanger F. Sequences, sequences, and sequences //Annual review of biochemistry. – 1988. – Т. 57. – №. 1. – С. 1-29.
2. Halberstam H., Roth K. F. Sequences. – Springer Science & Business Media, 2012.
3. Sacks H. On the preferences for agreement and contiguity in sequences in conversation //Language in Use. – Routledge, 2020. – С. 8-22.
4. Gauthier J. et al. A brief history of bioinformatics //Briefings in bioinformatics. – 2019. – Т. 20. – №. 6. – С. 1981-1996.
5. Zou Q. et al. Sequence clustering in bioinformatics: an empirical study //Briefings in bioinformatics. – 2020. – Т. 21. – №. 1. – С. 1-10.
6. Раев В.К., Цветков В.Я. Логические цепочки // Дистанционное и виртуальное обучение. 2018. - № 1(120). – с.14-21.
7. Liu Z. et al. Optimization of fuzzy demand distribution supply chain using modified sequence quadratic programming approach //Journal of Intelligent & Fuzzy Systems. – 2019. – Т. 36. – №. 6. – С. 6167-6180.
8. Chan W. et al. Imputer: Sequence modelling via imputation and dynamic programming //International Conference on Machine Learning. – PMLR, 2020. – С. 1403-1413.
9. Математика. Большой энциклопедический словарь / Гл. ред. Ю.В. Прохоров. 3-е изд. - М.: Большая Российская энциклопедия, 2000. 848 с.
10. Hebel M. et al. Sequence programming with dynamic boronic acid/catechol binary codes //Journal of the American Chemical Society. – 2019. – Т. 141. – №. 36. – С. 14026-14031.
11. Thomas L. et al. Learning styles and performance in the introductory programming sequence //ACM SIGCSE Bulletin. – 2002. – Т. 34. – №. 1. – С. 33-37.
12. Floyd R. W. Nondeterministic algorithms //Journal of the ACM (JACM). – 1967. – Т. 14. – №. 4. – С. 636-644..
13. Цветков В.Я. Систематика информационных ситуаций // Перспективы науки и образования. - 2016. - №5 (23). - С.64-68.
14. Титов Е.К. Алгоритмы ситуационной обработка информации // Славянский форум. -2018. – 4 (22). - С. 60-64.
15. Цветков В.Я. Применение принципа субсидиарности в информационной экономике // Финансовый бизнес. -2012. - №6. - С.40-43.
16. Козлов А. В. Анализ субсидиарных систем // Вестник Рязанского государственного радиотехнического университета. 2019. № 69. С.160-167.

17. Tsvetkov V.Ya. Application of Monte Carlo Method for Calculation of the Comets' Area by the Photographic Pictures // Modeling of Artificial Intelligence. 2017, 4(2): 96-101.
18. Duntemann J. Assembly language step-by-step: Programming with Linux. – John Wiley & Sons, 2011.

References

1. Sanger F. Sequences, sequences, and sequences //Annual review of biochemistry. – 1988. – T. 57. – №. 1. – C. 1-29.
2. Halberstam H., Roth K. F. Sequences. – Springer Science & Business Media, 2012, 34 c.
3. Sacks H. On the preferences for agreement and contiguity in sequences in conversation //Language in Use. – Routledge, 2020. – C. 8-22.
4. Gauthier J. et al. A brief history of bioinformatics //Briefings in bioinformatics. – 2019. – T. 20. – №. 6. – C. 1981-1996..
5. Zou Q. et al. Sequence clustering in bioinformatics: an empirical study //Briefings in bioinformatics. – 2020. – T. 21. – №. 1. – C. 1-10.
6. Raev V.K., Tsvetkov V.YA. Logicheskie cepochki // Distancionnoe i virtual'noe obuchenie. 2018. - № 1(120). – C.14-21.
7. Liu Z. et al. Optimization of fuzzy demand distribution supply chain using modified sequence quadratic programming approach //Journal of Intelligent & Fuzzy Systems. – 2019. – T. 36. – №. 6. – C. 6167-6180.
8. Chan W. et al. Imputer: Sequence modelling via imputation and dynamic programming //International Conference on Machine Learning. – PMLR, 2020. – C. 1403-1413.
9. Matematika. Bol'shoj enciklopedicheskij slovar' / Gl. red. YU.V. Prohorov. 3-e izd. - M.: Bol'shaya Rossijskaya enciklopediya, 2000. 848 s.
10. Hebel M. et al. Sequence programming with dynamic boronic acid/catechol binary codes //Journal of the American Chemical Society. – 2019. – T. 141. – №. 36. – C. 14026-14031.
11. Thomas L. et al. Learning styles and performance in the introductory programming sequence //ACM SIGCSE Bulletin. – 2002. – T. 34. – №. 1. – C. 33-37.
12. Floyd R. W. Nondeterministic algorithms //Journal of the ACM (JACM). – 1967. – T. 14. – №. 4. – C. 636-644.
13. Tsvetkov V.YA. Sistematika informacionnyh situacij // Perspektivy nauki i obrazovaniya. - 2016. - №5 (23). - C.64-68.
14. Titov E.K. Algoritmy situacionnoj obrabotka informacii // Slavyanskij forum. -2018. – 4 (22). - C.60-64.
15. Tsvetkov V.YA. Primenenie principa subsidiarnosti v informacionnoj ekonomike // Finansovyj biznes. -2012. - №6. - C.40-43.
16. Kozlov A. V. Analiz subsidiarnyh sistem // Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta. 2019. № 69. C.160-167.
17. Tsvetkov V.Ya. Application of Monte Carlo Method for Calculation of the Comets' Area by the Photographic Pictures // Modeling of Artificial Intelligence. 2017, 4(2). C.96-101.
18. Duntemann J. Assembly language step-by-step: Programming with Linux. – John Wiley & Sons, 2011.