УДК 004.021

ОБ УПРАВЛЕНИИ ВЫЧИСЛИТЕЛЬНЫМИ ПРОЦЕССАМИ

Д. т. н. О. Н. Паулин, к. т. н. Н. О. Комлевая, к. т. н. С. Ю. Марулин

Одесский национальный политехнический университет
Украина, г. Одесса
paolenic@yandex.ua, nokoml@yandex.ua, stasfoot@mail.ru

Рассматривается управляющая составляющая организации вычислительных процессов, которая опирается на теорему Бёма-Якопини о простейших структурах управления. Описываются три варианта организации цикла: for, while…do, do…while. Приводятся примеры иерархического построения известных вычислительных процессов.

Ключевые слова: вычислительный процесс, теорема Бёма-Якопини, структура управления.

Несоблюдение требований, предъявляемых к современным технологиям разработки программного обеспечения (ISO 12207, ISO 9000, CMM), приводит к большим затратам времени и ресурсов на исправление ошибок и реинжиниринг в будущем. Наиболее критическими являются этапы программирования и тестирования [1], которые напрямую влияют на качество ПО (ISO 9001). Поэтому качественное программирование и тестирование программ позволит значительно сократить время на разработку и повысить качество ПО в целом.

В последнее время популярным стало автоматное программирование [2], которое позволяет значительно сократить время на тестирование программ. Однако этому подходу присуща недостаточная вычислительная мощность. Альтернативой ему может послужить б***о***льшая выразительная мощность сетей Петри [3], которая занимает промежуточное положение между конечными автоматами и машиной Тьюринга.

Целью работы является снижение времени анализа программ за счет выделения макроопераций в вычислительных процессах.

В [4] рассмотрено первое приближение к представлению вычислительных процессов (ВП) сетями Петри и введено понятие «макрооперация», под которой понимается логически законченный фрагмент ВП. ВП содержит две составляющие: чисто вычислительную и управляющую. В продолжение [4] ниже рассматривается управляющая составляющая ВП.

В теоретическом плане схема алгоритмов – это ориентированный граф, содержащий вершины трех типов [5]:

- *функциональная* (используется для представления функции f: X→Y, т. е. арифметического выражения);
- *предикатная* (используется для представления предиката p: X→{FALSE, TRUE}, т. е. логического выражения, которое определяет управление по одной из возможных ветвей);
- *объединяющая* (реализует передачу управления от одной из двух входных ветвей к одной выходной ветви).

Из данных вершин графа составляются простейшие *структуры управления* вычислительным процессом: 1) *композиция* (следование), которая записывается в виде S_1; S_2; 2) *альтернатива* (выбор), которая записывается в виде **if** C **then** S_1 **else** S_2; в частном случае при $S_2=\varnothing$ получается часто используемый сокращенный оператор **if** C **then** S_1; 3) *итерация* (*цикл*), которая записывается как **for**, **do…while**, **while… do**. Здесь C (condition) интерпретируется как булево выражение, а S_1 и S_2 – как операторы (процедуры).

Согласно теореме Бёма–Якопини [6], любой исполняемый алгоритм может быть преобразован к структурированному виду, то есть к такому виду, когда ход его выполнения определяется только при помощи трех структур управления: последовательной, ветвлений и циклов.

Цикл – это многократное повторение некоторой совокупности операторов (процедур), называемой *телом* цикла; существует три варианта организации циклов:

– *счетный* – задано количество итераций (*for*);

– *условный 1-го рода* – окончание цикла определяется предусловием (*while…do*);

– *условный 2-го рода* – окончание цикла определяется постусловием (*do…while*).

Формулы ВП для данных циклов имеют соответственно вид: $I^i_m(T)$, $I^q(F; T)$ и $I^q(T; F)$. Здесь I^i_m – множество итераций по параметру i, $1 \leq i \leq m$; I^q – множество итераций по параметру q; F – выбор по условию; T – тело цикла.

Отметим, что вопрос функциональной полноты предложенных структур в [6] не ставился, а формулировка теоремы и термин «структура управления» (СУ) появляются в более поздних работах, например, в [5]. Однако и здесь в явном виде не говорится о функциональной полноте данных СУ.

Отметим, что цикл является синтетической СУ. Поэтому в данной работе оставлены первые две СУ (композиция и альтернатива), но в качестве третьей СУ предложено использовать петлю обратной связи, заданной оператором **goto**, что может быть записано в виде **r: S; goto r**; . Тогда набор СУ будет обладать следующими свойствами: неделимость каждой СУ; независимость СУ; минимальность набора. Этого набора достаточно для обеспечения функциональной полноты управления ВП.

Рассмотрим примеры конкретных иерархически построенных ВП.

Пример 1. ВП представляет собой сложение матриц: **C=A+B**. Формула данного ВП имеет вид $I^i_m(I^j_n(c_{ij}=a_{ij}+b_{ij}))$. Здесь иерархия (2 уровня) образована вложенностью цикла по j в цикл по i. Телом цикла является арифметическое выражение.

Пример 2. ВП – это процесс перемножения матриц: **C=A**x**B**. Формула данного ВП имеет вид $I^i_m(I^j_n(\Sigma=0; I^k_l(c_{ij}=\Sigma a_{ik}*b_{kj}))$. Здесь Σ – символ суммирования с накоплением. Имеем трехуровневую организацию ВП за счет вложенности циклов. Телом цикла также является арифметическое выражение.

Пример 3. Формула фрагмента ВП имеет вид …; A; $I^q(F(p, r); T)$; B; r: C; … Здесь A, B, C – операторы (процедуры); p – условие (предикат); r – метка для перехода в случае p=1.

Рассмотрена организация управления ВП; а также вопрос о функциональной полноте структур управления. В приведенных примерах иерархического построения ВП структура управления образуется вложенностью фрагментов ВП, особенно сложной в случае циклов, которые включают в себя петли обратной связи. В дальнейшем предполагается переход к более крупным алгоритмическим фрагментам – к макрооперациям, для которых будут построены свои сети Петри.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Yang Y., He M., Li M., Wang Q., Boehm B. Phase distribution of software development effort // Proc. of the International Symposium on Empirical Software Engineering and Measurement (ESEM).– Kaiserslautern, Germany, 2008.– P. 384–392,

2. Любченко В. В. К проблеме создания модели параллельных вычислений // Труды Третьей международной конференции «Параллельные вычисления и задачи управления – PACO'2006».

3. Котов В. Е. Сети Петри.– Москва: Наука, 1984.

4. Паулин О. Н., Поляков Ю. С., Шульгин В. А. Представление вычислительных процессов сетями Петри // Труды интернет-конференции SWorld.– 2015.– Вып. 4(41), Т. 2.

5. Гудман С., Хидетниеми С. Введение в разработку и анализ алгоритмов.– Москва: Мир, 1981.

6. Böhm Corrado, Jacopini Giuseppe. Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules. Communications of the ACM, May 1966, vol. 9, iss. 5. P. 366–371.

O. N. Paulin, N. O. Komlevaya, S. U. Marulin

On computational processes control

The authors consider the control component of computational processes organization based on Böhm–Jacopini structured program theorem. The paper also describes three variants of cycle organization: for, while ... do, do ... while. The examples of hierarchical construction of known computational processes are given.

Keywords: *computational process, Böhm-Jacopini theorem, control structure.*