

4. Коротких Ю.Г. Математическая модель термопластичности при больших упругопластических деформациях//Методы решения задач упругости и пластичности: Межвуз. сб.; Горьк. ун-т. Горький, 1973. Вып.7. С. 17-27.

УДК 681.3.06

В.В.Сидельников, А.В.Соницев, В.Л.Хомченко

### ФОРМАЛЬНОЕ ПРЕДСТАВЛЕНИЕ ФУНКЦИОНАЛЬНОГО НАПОЛНЕНИЯ ДИАЛОГОВОЙ СИСТЕМЫ УПРАВЛЕНИЯ ППП

(г. Ленинград)

Использование ЭВМ в любом виде человеческой деятельности, включая научные исследования, предполагает наличие некоторого формального способа описания решаемых задач, для которого существует "эвристически эффективная стратегия" [4] преобразования его в программное обеспечение (Про), как некоторую специфическую форму выражения совокупного знания о решаемой задаче, "эпистемологически адекватную" (там же) данному описанию. Таким образом, при решении задач с использованием средств вычислительной техники имеет место двухуровневая организация их представления: на верхнем - формальное описание решаемой задачи; на нижнем - Про, реализующее автоматическое решение данной задачи.

Предлагается формализованное описание исследовательских задач, ориентированное на использование ЭВМ в процессе поиска их решения. В основу описания Про положен модульный принцип его построения.

Язык описания исследовательских задач. Для описания задач исследования предлагается использовать следующие синтаксические конструкции.

Если для решения задачи  $A$  необходимо решить в заданной последовательности задачи  $B$ ,  $C$  и т.д., то задача  $A$  описывается в виде  $A: B, C \{, D\}$

Здесь и далее употребление метаскобок  $\{$  и  $\}$  соответствует повторению конструкции. Если, например, для решения некоторой задачи  $S$

необходимо:  $A$  - провести измерения;  $B$  - обработать полученные результаты;  $C$  - представить результаты в заданной форме, то задача  $S$  описывается в виде  $S: A, B, C$ .

Если для решения задачи  $A$  достаточно решить одну из задач: либо  $B$ , либо  $C$  и т.д., то задача  $A$  описывается в виде  $A: \text{или } B, C, \{D\}$ .

Например, задача  $A$  может состоять в вычислении математического ожидания измеряемого параметра, тогда  $B$ ,  $C$  и т.д. будут соответствовать различным методам получения требуемого результата.

Если для решения задачи  $A$  необходимо многократно решать ее подзадачу  $B$ , то задача  $A$  описывается в виде

$A: \text{цикл}(B)$

Например, если задача  $A$  представляет собой задачу наблюдения за поведением объекта, то задача  $B$  - проведение однократного измерения, или если  $A$  - задача выбора критерия качества функционирования некоторого объекта, то  $B$  - задача построения модели этого объекта и т.д.

Следует отметить, что такое описание исследовательских задач определенным образом отражает методологический прием, используемый при решении любых "интеллектуальных задач" [3] человеком, когда для решения исходной задачи намечаются альтернативные пути ее решения, последовательность действий, приводящая к искомому решению, наконец, выделяется подзадача, повторяющееся решение которой должно позволить решить требуемую задачу. Последняя ситуация имеет место всегда, когда для решения задачи исследования планируется проведение серии экспериментов. Кроме того, именно возможность представления описанным образом задачи создания программы для ЭВМ лежит в основе применения для этих целей структурного программирования.

Использование двух первых синтаксических конструкций при описании исследовательских задач семантически эквивалентно представлению интеллектуальных задач в искусственном интеллекте, которое "предусматривает разбиение исходной задачи на множество подзадач" [5] и описывается в виде графа редукции. Первая синтаксическая конструкция соответствует И-структуре взаимосвязи задач и определяет конъюнктивную вершину И-ИЛИ графа, вторая - ИЛИ-структуру и определяет дизъюнктивную вершину. Пример расширенного графа редукции путей введения дополнительной циклической структуры взаимосвязи задач представлен на рисунке. Этому описанию представления в виде графа

соответствует следующее описание с использованием предложенных синтаксических конструкций

$A: \text{выб}(B, C)$

$B: D, E$

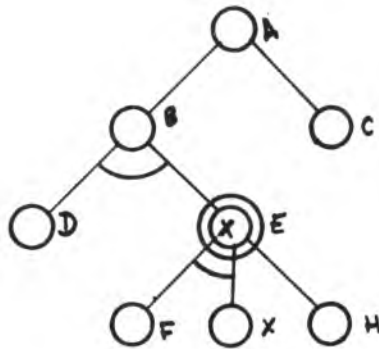
$E: \text{цикл}(X)$

$X: \text{выб}((F, X), H)$

Описание этой же задачи в более компактной форме, для которого произведены все возможные подстановки, будет иметь вид

$A: \text{выб}((D, \text{цикл}(X)), C)$

$X: \text{выб}((F, X), H)$



Р и с. Пример расширенного графа редукции исследовательской задачи

Аналогом описания представления в виде приведенного И-ИЛИ-графа является описание, в которое введены дополнительные подзадачи. В данном примере - это подзадача  $Y$ . Приведенное описание имеет вид

$A: \text{выб}(B, C)$

$B: D, E$

$E: \text{цикл}(X)$

$X: \text{выб}(Y, H)$

$Y: F, X$

Задачу, имя которой встречается только в левой части описания, назовем исходной задачей исследования. Задачи, имена которых встречаются только в правой части - элементарными подзадачами. Элементарная подзадача - это такая задача, решение которой априори известно исследователю. Каждой такой задаче может и должен соответствовать определенный модуль, реализованный либо аппаратно, либо программно. Выделение таких модулей в составе Про исследования определяет его "модульную архитектуру" [1].

Поскольку исследователь при описании своей задачи фактически строит граф редукции, в нем возможно появление тупиковых вершин. Тупиковым вершинам соответствуют задачи, которые не могут быть решены в рамках их представления исследователем. Путь, начинающийся в корневой вершине и заканчивающийся в тупиковой, либо должен быть удален из описания задачи, либо тупиковая вершина должна быть

"превращена" в конечную путем привлечения профессиональных программистов или дополнительных знаний о решаемой задаче. В любом случае "разрешение" тупиковой вершины основывается на аналогичном описании данной задачи, если для этого используются методологические приемы структурного программирования. Элементарными подзадачами при этом могут являться либо команды процессора, либо операторы присвоения языка высокого уровня, либо программы библиотек, либо, наконец, специализированные пакеты прикладных программ (ППП). В дальнейшем будем считать, что тупиковые вершины в расширенном графе редукции отсутствуют, он задан в явном виде и определяет конкретный этап исследований.

Программирование исследований. Для решения каждой отдельной подзадачи, не являющейся элементарной, естественно представляется применение "алгоритма перебора в глубину". В этом случае переход к решению очередной конъюнктивно связанной подзадачи осуществляется только после подучения решения предыдущей подзадачи, т.е. тогда, когда в процессе обхода графа достигнуты ее конечные вершины. Именно так осуществляется процесс исследования - для решения задачи, состоящей из последовательности подзадач, сначала решается каждая из них. Введение порядка обхода вершин графа означает, что описание задачи можно трактовать как план ее решения или "план действий, подлежащий выполнению некоторым исполнителем", в данном случае исследователем. Такой план называется программой [2].

Рассмотрим выполнение программы, соответствующей приведенному графу редукции, при этом каждая строка этой программы будет называться оператором. Первым выполняемым оператором будет оператор  $A: B \vee C (B, C)$ , так как имя задачи  $A$  не встречается в правых частях других операторов этой программы. Исследователю задается вопрос: "Что делать?  $B$  или  $C$  ?". Если он выберет  $C$ , выполнится соответствующий модуль, и решение задачи закончится. Модуль  $C$ , например, может осуществлять распечатку всех действий исследователя. Если будет выбрано  $B$ , то сначала выполнится модуль  $D$ . Если бы задача  $D$  была не элементарной, то следующим выполнялся бы оператор программы, у которого идентификатор  $D$  записан в левой части. Поиск решения задачи продолжался бы до тех пор, пока не были бы достигнуты соответствующие конечные вершины И-ИЛИ-графа. В данном случае задача  $D$  элементарная, и ее решение может быть получено без участия исследователя путем активизации и выполнения соответст-

вующего модуля. Решение задачи  $E$  начинается с вопроса к исследователю: "Надо ли решать задачу  $X$  ?". Если исследователь ответит: "Да", то ему будет задан вопрос: "Что делать?" Решать задачу  $Y$  или  $H$  ?". Если исследователь выберет  $Y$ , то выполнится модуль  $F$ , после чего будет задан вновь предыдущий вопрос. Так будет продолжаться до тех пор, пока не будет получен ответ: "Решать задачу  $H$ ". После выполнения соответствующего модуля будет выполняться оператор  $E$ ; ЦИКЛ( $X$ ), и поиск решения будет продолжен так, как это описано выше, до тех пор, пока от исследователя не будет получен ответ о том, что задачу  $X$  больше решать не требуется. Поскольку у вершины  $B$  нет больше конъюнктивно подчиненных вершин, процесс поиска решения будет закончен.

Необходимость использования двух различных способов определения повторяющихся действий исследователя связана с организацией данных, используемых при решении соответствующей задачи. Будем считать, что все данные записываются в общую память в порядке их возникновения и что очередная итерация, соответствующая решению задачи, описанной внутри тела оператора цикла, должна начинаться при тех же значениях исходных данных, что и любая предыдущая, а при использовании рекурсии данные могут накапливаться. Приведем пример. Пусть имеется модуль "измерение". Если необходимо просто измерять и предъявлять результаты исследователю, то для описания такой задачи следует использовать оператор цикла. Если предполагается вести измерения с целью накопления измеряемых параметров, то следует использовать рекурсивное описание задачи "вести измерения".

Язык программирования исследований. Предложенный язык описания исследовательских задач отличается от функциональных языков программирования тем, что определение значений функций управляющих программой производится не автоматически, а путем обращения к исследователю. В предлагаемом языке зафиксированы два типа таких функций "выб" и "цикл". Областью значений функций первого типа является множество целых чисел от единицы до числа задаваемых альтернатив, второго "истина" и "ложь". Для образования имен функций предлагается использовать обозначения их типа, префиксы и суффиксы. Префиксы определяют вычислительную обстановку, в которой производится определение значений функций: до начала счета, в процессе счета, по прерыванию от клавиатуры исследователя. Суффиксы задают правила образования текста вопроса и словарь, из слов которого формируется вопрос. а также правила вычисления значений функций в соответствии

с полученным ответом. Если диалог ведется в жестком формате, то каждому суффиксу достаточно поставить в соответствие фрагмент обшего текстового файла всех вопросов.

Предварительная настройка терминологии диалога обеспечивается введением операторов вида

*выбтер: выбор(выбт1, выбт2{, выбт3})*

*циклтер: выбор(циклт1, циклт2{, циклт3})*,

где *терм, оп, т1, т2* и т.д. - любые суффиксы, задаваемые исследователем.

Технология проектирования ПрО исследований. Для создания ПрО научных исследований текст программы на описанном языке транслируется в персональный пакет прикладных программ (ППП), названный так потому, что он создается исследователем самостоятельно. По тексту программ определяются имена элементарных подзадач. Для их решения исследователем формируется библиотека модулей и описываются типы данных, используемые в каждом модуле. На основе морфологического разбора имен функций, управляющих программой, производится заполнение соответствующих текстовых файлов. Создание ПрО осуществляется в диалоговом режиме и может быть описано в виде программы на предложенном языке. В дальнейшем работа с ППП осуществляется так, как это описано выше.

#### Библиографический список

1. Ершов А.П., Ильин В.П. Пакеты программ - технология решения прикладных задач.-Новосибирск:ВЦ СО АН СССР, 1978.- 22 с.
2. Ершов А.П. Программа //Математическая энциклопедия. М.:Советская энциклопедия. Т.4.-С.645.
3. Ефимов Е.И. Решатели интеллектуальных задач.-М.:Наука, 1983.- 320 с.
4. Попов Э.В., Фридман Г.Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта.-М.:Наука, 1976.-456 с.
5. Робототехнические и гибкие автоматизированные производства В 9 кн. Кн.6. Назаретов В.М., Ким Д.П. Техническая имитация интеллекта.-М.:Высшая школа, 1986.- 144 с.