Э. М. ПОГОСЯН, А. В. КАЛАНТАРЯН, Б. К. КАРАПЕТЯН

ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ ИССЛЕДОВАНИЙ МЕТОДОВ АДАПТАЦИИ КОМБИНАТОРНЫХ АЛГОРИТМОВ УПРАВЛЕНИЯ

1. Проблема адаптации комбинаторных алгоритмов управления (АКА-У) состоит в построении систематических итеративных методов синтеза решений комбинаторных проблем управления, в частности, проблем поиска оптимальных стратегий в позиционных играх [1]. Соответствующие проблемно-ориентированные пакеты прикладных программ (ППП) предназначены для автоматизации определенной части исследований указанной проблемы [1, 2]. Комплектация пакетов программами должна обеспечить возможность доказательства (опровержения) гипотез, связанных с АКА-У для заранее выбранной проблемы-посредника с последующим их обобщением на проблемы класса, представленного этим посредником.

В разрабатываемом пакете АКА-У3 в качестве проблемы-посредника, так же как и в описанных в [1, 2] пакетах АКА-У1, У2, выбрана проблема поиска оптимальных стратегий в шахматах (ПОС^{ш}).

2. Основные функциональные требования, предъявляемые к пакету, следующие:

2.1. обеспечение возможности измерения с определенной точностью эффективности алгоритмов поиска стратегий,

2.2. обеспечение возможности компактного и близкого шахматистам описания классов объектов, связанных с ПОС^{ш} в частности, позиций, стратегий, партий и др.,

2.3. обеспечение достаточно представительного фонда записей источников, из которых шахматист способен извлекать знания; в их числе записей партий и их комментариев, учебных текстов, эндшпильных и дебютных справочных,

2.4. обеспечение оперативного поиска и извлечения записей из фондов по запросам, близким запросам шахматистов, а также необходимого пополнения фондов.

2.5. наличие известных алгоритмов поиска стратегий и их компонент; в частности компонент планирования в графе целей, оценки правдоподобия целей, синтеза стратегий и др.,

2.6. наличие известных локальных операторов усиления алгоритмов поиска стратегий,

2.7. наличие известных процедур синтеза закономерностей и их компонент.

Формулируемые ниже технические требования к пакету АКА-У3 получены в результате анализа опыта построения и эксплуатации пакетов АКА-У1-У2 и направлены к устранению обнаруженных при этом недостатков. Рассмотрим основные из них.

3.1. Пакет АКА, У1 был завершен к 1979 г. для ЭВМ Минск-32, имел объем 25000 операторов на языках ЯСК (12000 операторов) и Фортран (13000 операторов). Основные недостатки этого пакета относились к (а) малой скорости единичных экспериментов на Минск-32, (б) низкому уровню и неприспособленности языков ЯСК и Фортран для реализации алгоритмов поиска стратегий, (в) недостаточной модульности дескрипторов и, как следствие, дублированию построения функционально эквивалентных программ.

С учетом указанных трудностей построение пакета АКА-У2 проводилось на ЭВМ ЕС1030 и языке высокого уровня ПЛ/1. Возможности пакета были экспериментально проверены при реализации алгоритма поиска стратегий, являющегося по существу модификацией идей Ж. Питра [3]. Основные трудности реализации в целом были обусловлены недостаточно удобной символикой и громоздкостью языка ПЛ/1, созданного в 60-е годы. Рассмотрим некоторые из них.

3.2.1. Неудачна система соглашений о связях между отдельными модулями. Правила ПЛ/1 позволяют описывать интерфейс не полностью или вовсе опускать описание. В этих случаях в силу вступают определенные соглашения, которые не всегда устраивают программиста. Громоздкость описания всех связей делает программу трудной для восприятия, а практика умолчаний может привести к трудно обнаруживаемым ошибкам. Существует ряд методов с привлечением препроцессора ПЛ/1, позволяющих как-то упорядочить процесс описания интерфейса, но все они недостаточно эффективны. На сегодняшний день основной процент ошибок в системах, написанных на ПЛ/1, приходится на межмодульные связи.

3.2.2. В синтаксисе языка ПЛ/1 имеется некоторое ограниченное число типов переменных. В более поздних языках программирования обычно имеются средства для определения специальных типов переменных, обеспечивающих более удобный и безопасный доступ к данным, чем в ПЛ/1. Например, тип переменной «натуральное число» не может принимать отрицательные значения, и при любой попытке нарушить это условие процесс вычислений будет прерван. Наиболее мощным средством, отсутствующим в ПЛ/1, является возможность описания типов переменных с не только числовыми значениями. Например, переменная типа «цвет» может принимать значения типа «белый» или «черный», а не 0 или 1 соответственно. Тогда стало бы некорректным присваивание переменной типа «цвет» значения переменной типа «размер». Ошибки такого рода были бы выявлены еще на стадии компиляции.

3.2.3. При написании системы программ на ПЛ/1 количество внешних (или глобальных) переменных обычно велико, причем любой мо-

дуль имеет неограниченный доступ к каждой переменной. Это создает значительные трудности на всех этапах создания системы. Некоторые языки высокого уровня позволяют организовать защиту глобальных переменных от случайной порчи. ПЛ/1 не справляется с этой задачей. Более того, неправильное использование операции DEFINE может привести к настолько непредсказуемым результатам, что бывает целесообразнее переписать систему заново, чем искать причину ошибок.

3.2.4. Управляющие структуры ПЛ/1 недостаточно гибки. Отсутствуют CASE-структура, группа DO-UNTIL, операция ELSE-IF, возможность покидать тело цикла, не нарушая правил структурного программирования и так далее. Моделирование таких структур средствами ПЛ/1 без привлечения оператора GOTO делает программу более громоздкой и ухудшает ее наглядность.

3.2.5. Невозможность организации динамической структуры системы средствами ПЛ/1 ОС ЕС. Для систем со значительным количеством составляющих модулей это очень серьезное ограничение.

3.2.6. Многословная и неудачная диагностика стандартного компилятора ПЛ/1 ОС ЕС. Обилие сообщений приводит к тому, что их редко просматривают полностью. Возможность опускать описания переменных, заимствованная из ФОРТРАНа, является источником множества ошибок. При опечатке или сбое вводного устройства текст программы искажается, а компилятор вместо выдачи диагностического сообщения может полностью нарушить логику программы.

3.2.7. Реализация программ, полученных посредством стандартного компилятора ПЛ/1 ОС ЕС, неэкономична как с точки зрения расхода памяти, так и по быстродействию. Отсутствуют такие полезные возможности как сбор статистики по используемости отдельных модулей системы (так называемое профилирование).

3.2.8. Вызов процедур реализуется настолько неэкономично, что для получения хороших временных характеристик минимизируют количество модулей. При этом ухудшается как наглядность программ, так и время на их написание и отладку.

3.3. При создании пакета АКА-У1 были проведены работы по машинному представлению и использованию некоторых шахматных понятий, задаваемых языком первого порядка. С этой целью были выделены ряд шахматных понятий, получены их формульные представления в языке первого порядка, для каждой формулы составлены фортран-программы (дескрипторы формул), проверяющие ее выполнимость в заданной позиции [1, 2].

Предполагалось, что будет выделено некоторое множество элементарных дескрипторов, из которых будут скомпанованы более сложные. При реализации этого подхода на языке ФОРТРАН на ЭВМ Минск-32 были выявлены недостатки двух типов:

I. Связанные со слабой структурной организацией фонда дескрипторов;

II. Связанные с несоответствием языка ФОРТРАН решаемой задаче. К недостаткам I типа относятся:

3.3.1. Включение в множество элементарных дескрипторов понятий разной степени сложности. Например, чисто статическое понятие «центр доски», находится в том же классе, что и понятие «количество ударов на клетку», в котором необходимо учитывать взаиморасположение фигур в заданной позиции.

3.3.2. Нестандартный интерфейс программа — подпрограмма. Различная структура входных и выходных параметров дескрипторов одного уровня затрудняет контроль над значениями параметров, откладку и модификацию программ.

3.3.3. Использование больших COMMON—блоков для передачи параметров. Это приводит к многочисленным побочным эффектам при добавлении новых дескрипторов, усложняет сопровождение и модификацию фонда дескрипторов.

3.3.4. Невозможность расширения фонда дескрипторов без программирования. Добавление нового понятия, даже если оно определяется с помощью дескрипторов фонда, предполагает написание новой программы, что нетривиально по указанным выше причинам.

3.3.5. Ограниченность области применения дескрипторов. Использование некоторого дескриптора вне фонда связано со значительными программными изменениями.

Недостатки II типа следующие:

3.3.6. Отсутствие развитых структур управления, что приводит к увеличению объема программ и сложности тестирования;

3.3.7. Отсутствие средств символьной обработки. Последнее усложняет представление формул, приводит к необходимости кодирования/декодирования, увеличивает используемые объемные и временные ресурсы.

3.3.8. Отсутствие возможности определить новые типы данных и операций над ними, более соответствующих структуре используемых объектов (клетка, позиция, цвет, фигура, линия и т. д.) и позволяющих оперировать с ними как с первичными данными.

4.1. Вышеизложенное позволяет сформулировать следующие технические требования к выбору языка программирования, построению транслятора и структурной организации фонда дескрипторов поиска АКА-У3.

4.2. Требования по выбору языка программирования и построению транслятора:

4.2.1. Автоматическая верификация межмодульного интерфейса на стадии компиляции.

4.2.2. Допустимость нечисловых типов переменных.

4.2.3. Ограничение доступа к глобальным переменным, а именно полный запрет при отсутствии разрешения, и разрешение только на чтение. Для получения разрешения на запись должен быть специальный оператор, находящийся в начале модуля.

4.2.4. Наличие широкого ассортимента управляющих структур языка, включая операторы CASE, DO-UNTIL, LEAVE (для покидания тела цикла).

4.2.5. Возможность организации как оверлейных, так и динамических структур загрузочных модулей средствами языка высокого уровня.

4.2.6. Лаконичная и по возможности полная диагностика на стадии компиляции.

4.2.7. Хорошие эксплуатационные характеристики объектных модулей, порождаемых транслятором (имеются в виду быстродействие и объем памяти).

4.2.8. Оптимальный (по быстродействию) механизм вызова процедур.

4.3. Требования к структурной организации фонда дескрипторов:

4.3.1. Вычисление дескриптора ресурсами специальной программной компоненты (пакета в смысле языка АДА), для которой формула каждого дескриптора, наряду с заданной позицией, является входным параметром.

4.3.2. Возможность представления и обработки дескрипторов непосредственно в терминах языка первого порядка (без дополнительного кодирования).

4.3.3. Возможность обработки замкнутых формул с ограниченными кванторами.

4.3.4. Возможность определения типов данных с явным указанием их области значений для задания используемых элементов универсума.

4.3.5. Стандартизация представления используемых функций языка первого порядка, их входных/выходных параметров и способов вычисления.

4.3.6. Возможность простой модификации и расширения набора используемых функций языка первого порядка.

4.3.7. Возможность вычисления любой правильно построенной формулы языка первого порядка, если функции, используемые в ней, включены в систему (без необходимости программирования).

4.3.8. Возможность использования компоненты вычисления дескрипторов вне зависимости от контекста программной среды (например, как процедуры функции).

5.1. В настоящее время при выборе основного языка программирования пакета АКА-УЗ предпочтение отдано языку АДА [4]. Благодаря более совершенному синтаксису язык АДА полностью обходит первые четыре (4.2.1.—4.2.4.) из перечисленных затруднений. Требования 4.2.5—4.2.8 относятся скорее к реализации компилятора, чем к языку в целом. Представляется возможным создание компилятора с некоторого подмножества языка АДА. Компилятор должен по возможности быть лишен недостатков, характерных для компилятора ПЛ/1 ОС ЕС. Многие функции языка АДА, как, например, действия над вещественными числами, мультизадачность и некоторые другие для реализации пакета АКА-УЗ несущественны и могут быть опущены. По мере напи-

сания компонентов АКА-УЗ можно постепенно усиливать и компилятор. Для того, чтобы процесс работы не прерывался, необходима программная совместимость кода, порождаемого разными версиями компилятора.

5.2. В [5] описана реализация фонда дескрипторов, в основном удовлетворяющая требованию 4.3. С этой целью разработаны структуры данных для описания элементов дескрипторов, позволяющие систематизировать и унифицировать как представление дескрипторов, так и процедуры обработки их отдельных компонент. При этом обеспечивается выполнение требований 4.3.1.—4.3.3. Для выполнения требования 4.3.4 в языке ПЛ/1, на котором проверялись алгоритмы обработки дескрипторов, были использованы дополнительные программные модули моделирования определяемых типов (при использовании языка АДА это требование выполняется автоматически). Стандартизация функций, используемых для задания дескрипторов, достигнута введением объектов специального типа—«досок», что позволило обеспечить выполнение требований 4.3.5—4.3.8.

Է. Մ. ՊՈՂՈՍՅԱՆ, Ա. Վ. ՔԱԼԱՆԹԱՐՅԱՆ, Բ. Կ. ԿԱՐԱՊԵՏՅԱՆ

ՂԵԿԱՎԱՐՄԱՆ ԿՈՄԲԻՆԱՏՈՐ ԱԼԳՈՐԻԹՄՆԵՐԻ ԻՆՔՆԱՀԱՄԱԿԵՐՊՄԱՆ ՄԵԹՈԴՆԵՐԻ ՈՒՍՈՒՄՆԱՍԻՐՈՒԹՅԱՆ ԾՐԱԳՐԱՅԻՆ ԱՊԱՀՈՎՄԱՆԸ ՎԵՐԱԲԵՐՈՂ ՊԱՀԱՆՋՆԵՐԸ

Քննարկվում են վերնագրում նշված ծրագրային ապահովման ստեղծման հիմնական դժվարությունները PL-1 լեզուն օգտագործելու դեպքում։ Ձևակերպված են ծրագրային համակարգին ներկայացվող անհրաժեշտ ֆունկցիոնալ և համակարգային պահանջները և նշված են նրանց իրագործման ճանապարհները։

ЛИТЕРАТУРА

1. *Погосян Э. М.* Адаптация комбинаторных алгоритмов, Изд-во АН АрмССР, Ереван, 1983.
2. *Погосян Э. М., Амбарцумян М. А., Аджабян Н. А., Арутюнян Ю. Г., Джндоян Л. О.* Пакет прикладных программ для исследований методов адаптации комбинаторных алгоритмов управления (АКА-1, 2). Отчеты ВЦ АН АрмССР за 1974—1980 г.
3. *Pitrat J.* A Chess Combination Program which Uses Plans, Artificial Intelligence, 8, 275—321, 1977.
4. Язык программирования АДА (предварительное описание). Финансы и статистика, М., 1981.
5. *Карапетян Б. К.* Об одной унификации представления шахматных понятий. См. наст. сборник.