

ОБЪЕКТНО-ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ВЫЧИСЛЕНИЙ

Анд.Б.Климов

1. В современном процессе развития языков программирования доминируют две несовместимые модели вычислений: функциональная и объектно-ориентированная. В этом докладе предлагается "компромиссная" модель вычислений, наследующая у каждой из моделей определенные достоинства.

2. Функциональная модель вычислений (ФМВ) восходит к классической математической традиции. На ней основаны такие языки программирования, как Лисп, Рефал. Ее основные свойства: в определении используется теоретико-множественная денотационная семантика; области данных (называемые значениями) определяются как некоторые множества; конструкциям языка программирования "буквально" сопоставляются теоретико-множественные функции на зафиксированных областях данных.

В операционной семантике ФМВ имеется свобода выбора порядка вычисления: независимые функциональные термы (например, $G(X)$ и $H(Y)$ в $F(G(X), H(Y))$) могут вычисляться в любом порядке или, попеременно, "параллельно".

3. Объектно-ориентированная модель вычислений (ОМВ) - это "собственное изобретение" компьютерной науки, программирования. Она возникла в результате развития основной линии языков программирования от Фортрана через Алгол-60, Алгол-68, Симулу-67, Паскаль и другие до языка Ада. Характерные черты ОМВ: принци-

пильное использование операционной семантики, опирающейся на понятие последовательности вычислений ("процесса вычисления"); имеется новый вид данных — объекты, которые изменяются во времени вычисления; объекты идентифицируются ссылками, которые можно использовать в роли значений. Ссылки не являются "теоретико-множественными" данными, т.к. не существует множества всех ссылок. (Это и есть основное препятствие для совместимости двух моделей вычислений.) Ссылки возникают при выполнении операции "создать новый объект". Ссылка — результат этой операции обладает одним свойством: она не равна ни одной другой ссылке, имеющейся в данный момент вычисления. Равенство ссылок предоставляет понятие "тот же самый объект".

Универсум данных ОМБ образуется объединением некоторой теоретико-множественной области значений и ссылок на объекты и замыканием по операции "образования составного значения" (как минимум, "образования пары"). Например, расширяя символьную область данных языка Лисп (множество списков) или языка Рефал (множество выражений) естественно включить ссылки на права атомов Лиспа или символов Рефала. Универсум данных ОМБ не является множеством. Будем называть его элементы квазизначениями.

Каждый объект обладает состоянием, являющимся квазизначением. Результат операции над объектом зависит от ее аргументов, состояний объектов, доступных прямо или косвенно по ссылкам, переданным в аргументах, и, быть может, от другой информации о состоянии вычислительной системы. Операция над объектом может изменить его состояние и состояния доступных объектов. Заметим, что такое определение, как: "объект есть упорядоченная пара из ссылки и состояния" — можно использовать только как теоретико-множественную метафору, не имеющую точного смысла.

4. Из функционального языка программирования можно "сделать" объектно-ориентированный следующим образом:

- выбираем какой-нибудь порядок вычисления функциональных термов (обычно "слева направо" и "изнутри наружу");
- вводим новый вид данных — ссылки на объекты;
- добавляем к первичным функциям языка какие-то операции над "объектами" (от их набора, конечно, зависит выразительная мощность языка, но здесь мы не будем касаться этого важного вопроса).

Ясно, что так преобразованный язык уже не является функциональным. Функции, определяемые на исходном функциональном языке,

превращаются в так называемые процедуры.

5. Реальные объектно-ориентированные языки (называемые также "процедурными", "императивными", "операторными"), конечно, буквально так не создавались. Они отличаются от функциональных не только семантическими понятиями, но и по синтаксису, "дизайну".

С другой стороны, все реализованные на ЭВМ так называемые "функциональные" языки являются таковыми только по синтаксису, а в действительности сделаны по приведенной схеме. Было бы правильнее называть их псевдофункциональными - "лже функциональными", так как никакими преимуществами, "хорошими" свойствами функциональных они не обладают. Их привлекательность только в наличии четко выраженного подязыка, обычно называемого "чисто функциональным" или просто "чистым", например, "чистый Лисп", "чистый Рефал".

6. Наша цель: определить квазифункциональный - "подобный функциональному" объектно-ориентированный язык.

Для этого выберем некоторые свойства функциональных языков в качестве "основных" и разрешим только такие первичные операции над объектами, при использовании которых эти свойства не нарушаются.

7. Для построения объектно-функциональной модели вычислений (ОФМВ) используем вариант ОМВ, допускающий так называемое "параллельное программирование", которое с логической точки зрения есть недетерминированное последовательное. Выбор последовательности вычислений нужен в ОМВ для определения результатов первичных операций над объектами. (Заметим, что семантика сомлок - их равенство - не требует порядка вычисления [1].) Этот выбор можно делать как до вычисления (последовательная ОМВ), так и случайно во время вычисления ("параллельная" или недетерминированная ОМВ).

Будем вычислять "параллельно" не вложенные друг в друга функциональные термы, например, $G(x)$ и $H(y)$ в $F(G(x), H(y))$. Это значит, что последовательности элементарных действий, составляющие в этом примере процессы вычисления $G(x)$ и $H(y)$, сливаются в одну последовательность любым способом (если только какая-либо из операций над объектами явно не требует противного, т.е. не "ожидает" изменения состояния какого-нибудь объекта).

Таким образом, конкретное вычисление в "параллельной" модели - всегда последовательное, но эта последовательность может

выбираться случайно. Это значит, что программа с исходными данными задает множество последовательностей вычисления.

8. Итак, назовем объектно-функциональной моделью вычислений (ОФМВ) недетерминированную объектно-ориентированную модель вычислений, у которой первичные операции над объектами таковы, что для любой программы выполняются следующие свойства, присущие чисто функциональным языкам:

- детерминированность результата (эквивиальность) - все допустимые варианты последовательности вычислений любого функционального термина дают эквивалентные результаты;

- повторная вычислимость - следующее преобразование процесса вычислений дает эквивалентный результат: в любой момент скопировать любой функциональный терм перед началом его вычисления, запустить "параллельное" вычисление обоих экземпляров и по окончании использовать любой из результатов.

9. Охарактеризуем класс процедур, удовлетворяющих этим свойствам, через следующую эталонную интерпретацию. Данные: объекты - это набор исходящих из вершины дуг и маркировка вершины и дуг значениями; этот набор дуг организован как множество кортежей дуг.

В результате вызова процедуры вырабатывается значение и/или подграф (кортеж новых вершин) и происходит "доопределение" состояний некоторых старых вершин: к ним добавляются кортежи исходных дуг. Два вызова процедуры с равными аргументами порождают конгруэнтные подграфы.

10. Предложенная объектно-функциональная модель вычислений (ОФМВ) предназначена для замены функциональной модели в теоретических исследованиях и в построении систем преобразований алгоритмов, в частности, в нее без каких-либо принципиальных осложнений переносятся известные методы частичных вычислений.

ОФМВ требует своего стиля программирования - не совсем обычного, но наполненного здравым смыслом. Этот стиль является продолжением функционального стиля программирования на задачи, использующие понятие объекта как модели "вещи", "предмета".

ОФМВ естественно реализуется на параллельных компьютерах. Свойство эквивиальности избавляет пользователя от проблем недетерминированного программирования в терминах взаимодействующих процессов. А свойство повторной вычислимости удобно не только для человека, но и "для компьютера": оно лежит в основе построения надежных систем, в которых при сбоях или отказе производится

пересчет потерянной информации.

Литература.

1. Анд.В.Климов. Использование концепции смешанных вычислений для конструирования объектно-ориентированного функционального языка программирования // Школа-семинар "Телави-83". - М.: ВИНТИ, 1983, с.67-70.

**СЕМИОТИЧЕСКИЕ АСПЕКТЫ
ФОРМАЛИЗАЦИИ
ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ**

**Всесоюзная школа-семинар
«Боржоми-88»**

ТЕЗИСЫ ДОКЛАДОВ И СООБЩЕНИЙ

МОСКВА 1988

Всесоюзный институт научной и технической информации

ГКИТ СССР и АН СССР

Институт кибернетики АН Грузинской ССР

Вычислительный центр АН СССР

**Научный совет по проблеме "Искусственный интеллект" Отделения
информатики, вычислительной техники и автоматизации АН СССР**

СЕМИОТИЧЕСКИЕ АСПЕКТЫ

ФОРМАЛИЗАЦИИ

ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Всесоюзная школа-семинар

г.Боржоми, 22-30 апреля 1988 г.

ТЕЗИСЫ ДОКЛАДОВ И СООБЩЕНИЙ



Москва 1988

Председатели Оргкомитета чл.-корр. АН ГССР Г.Д. х а р а т и ш в и л и
проф. П.В. Неостеров

Председатель Программного комитета академии Г.С. П о с п е л о в

Руководители секций

Проф. Д.А. П о с п е л о в , докт. филол.наук Вяч.В.И в а н о в

Составитель

к.с.и. В.К. Ф и н и

Редакторы

к.ф.-м.и. М.М. З а б е ж а й л о , Е.В. Р а х и л и н а ,
В.В. Р ы ж и н о в , к.ф.-м.и. Д.П. С и в о р ц о в ,
Е.Ф. Ф е б р и а н т о в а , М.В. Ф и л и п е н н о

Ответственный редактор

доц. Р.С. Г и я р е в о и я

С Е К Ц И Я I

ЛОГИЧЕСКИЕ ПРОБЛЕМЫ И ПРОГРАММНЫЕ СРЕДСТВА
ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Руководитель

Д.А. П о с п е л о в

Ученый секретарь

В.Ф. Ф а б р и к а н т о в а

	Сдано в набор 15,03,88	
Подписано в печать 26,02,88		Т-02869
Формат 60x90 1/16	Печать офсетная	Бум. офс.
Усл.печ.л 26,5	Усл.кр.-отт. 26,62	Уч.-изд.л 22,81
Тир. 250 экз.	Зах. 2226	Цена 1р.55к.

Производственно-издательский комбинат ВИНТИ
140010, Люберцы 10, Московской обл.,
Октябрьский проспект, 403