

# Аллегии как основа алгоритмов

Ричард С. Бёрд

Исследовательская группа в области программирования, Университет Оксфорда  
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK

## 1 Введение

Представьте исчисление, спроектированное как для расчёта отдельных алгоритмов, так и для изучения алгоритмических стратегий в общем; как оно должно выглядеть? Любопытно, что в то время как математический аппарат для анализа алгоритмов хорошо развит, включая правила обращения с биномиальными тождествами, рекуррентными соотношениями, производящими функциями, и т. д. (см., например, [9, 10]), разработка схожего инструментария для построения алгоритмов всё ещё считается мало важной многими проектировщиками алгоритмов. Вместо того, алгоритмические принципы, — такие как «разделяй и властвуй», метод ветвей и границ, и т. д. — продолжают объясняться на примерах, и отдельные алгоритмы уточняются на уровне формализации, намного более низком, чем, скажем, упрощение биномиальной формулы.

Исчисление для алгоритмов не совсем то же, что методология программирования, так как их цели различны. Мы стремимся не к теории создания надёжного программного обеспечения индустриального масштаба, а всего лишь к теории, допускающей выражение и рассуждение об алгоритмах на высоком уровне абстракции. Существует или должен существовать акцент на вычисления, т. е. эквивалентном или безэквивалентном рассуждении в формальных выражениях. Эквивалентная логика проста, и способность доказать что-либо при помощи вычислений это важный показатель зрелости математической теории. Кто-то может скептически отнестись к описанию деталей утомительных вычислений, или оставлять их только лишь для машины, но существование типового процесса вычислений это свидетельство успеха теории, а не её провала. Однако вопросы нотации важны, нужно иметь возможность проводить вычисления в лаконичных и прозрачных обозначениях, разработанных для человеческого, а не машинного восприятия.

Наша цель в этом докладе — показать, что одно подходящее исчисление для алгоритмов предоставляет теория аллегии. Аллегии, введённые Фрейдом и Сцедровым [8], описывают теоретико-категорное исчисление отношений. Исчисление отношений играет важную роль в исчислении алгоритмов, потому что само по себе исчисление функций не может предоставить необходимую степень свободы в спецификации и доказательстве. Теоретико-категорное исчисление подходит в силу, по меньшей мере, трёх взаимосвязанных причин. Во-первых, теория категорий предоставляет унифицированный каркас для алгоритмических идей; во-вторых, неоправданная избыточность в обозначениях снижается при использовании преимущественно бесточечного композиционного рассуждения в стрелках; в-третьих, теория категорий предоставляет простой инструмент, функтор, для лаконичного описания типов данных. В результате алгоритмические стратегии и решения могут быть сформулированы без упоминания конкретных типов данных. Специализированные алгоритмы для отдельных задач получаются подстановкой параметров функтора подходящим способом. Этот стиль порождения программ называется *обобщённым* или *много типовым* порождением программ (см., к примеру, [3, 12, 15]). Есть также некоторые свидетельства (см., например, [7]) того, что саму категорию полезно трактовать как параметр теории; подстановка различных категорий может тогда приводить к разным вычислительным парадигмам, например, функциональным, декларативным, основанным на состояниях или реактивным алгоритмам.

Наша цель в последующем изложении — дать обзор использования аллегорий в изучении алгоритмов. Понятия категории, функтора, естественного преобразования и т. д. считаются известными. Более полное изложение от базовых принципов опубликовано в [2]. Этот текст от части является сводкой результатов работы ряда исследователей (включая Блекхауза, Кокетта, де Мура, Фоккинги, Хаттона, Джея, Юринга, Малкольма, Миртенса, Мейера, Мёллера и других), преследовавших схожие цели. В частности, именно Блекхауз и де Мур были теми, кто первыми предложили использовать теорию отношений типов данных для конструирования программ, и де Мур — кто первый догадался применить аллегорическую основу.