К доказательному программированию для непрерывных данных

Н. Н. Непейвода (ИПС РАН, Переславль-Залесский)

Принципиальное различие

- **Дискретные объекты:** точные, конечные данные
- **Непрерывные объекты:** заданы приближениями, требуют работы с неполной информацией

Теоретическая база

Конструктивная математика (120 лет развития)

Ключевой вывод: конструктивные функции над хаусдорфовыми пространствами оказывались всегда непрерывнавсегда непрерывны, но не наоборот.

Фундаментальный результат

Непрерывность как вычислимость

Функция непрерывна $\Leftrightarrow$
она вычислима относительно некоторой целочисленной функции

Топологическая парадигма

В связи с этим в данном рассмотрении на первое место ставятся топологические аспекты вычислимости и их влияние на программирование. Не менее важные алгебраические и ресурсные аспекты не игнорируются но остаются на периферии.

Работа с конечной информацией

Конструктивная функция — это процесс уточнения

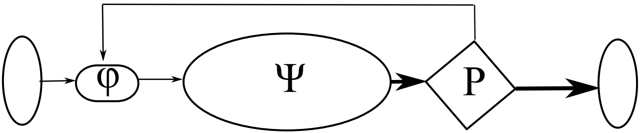

Рис.: Схема работы конструктивной функции

Окрестностное отношение

Показывает, какая точность аргумента нужна для получения заданной точности результата.
Важно: вычисляется не из функции, а из её программы (методами суперкомпиляции)

Модель вычислимости (Крайзель, Трулстра)

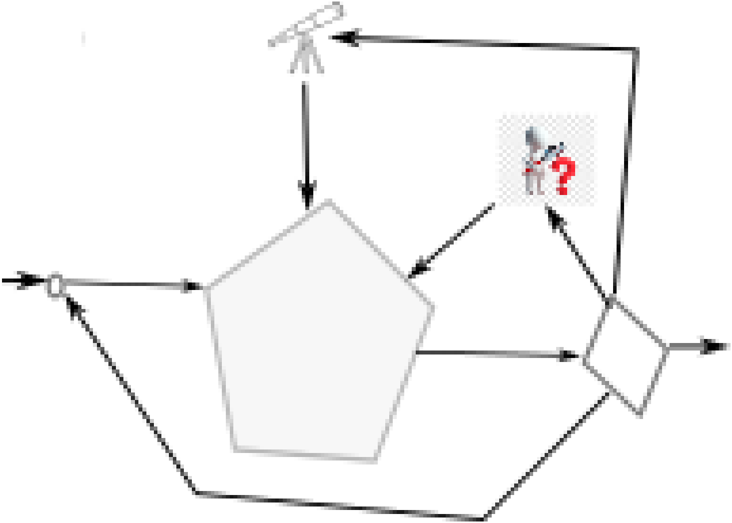

Рис.: Вычислительное устройство по Крайзелю и Трулстра

Пояснения

Три источника информации

1. **Дискретная информация:** известная, полная (программа, данные)
2. **Физический датчик:** беззаконная последовательность (известны только полученные данные)
3. **Творческая последовательность:** моделирует действия оператора

Вывод

Программы для непрерывных объектов — это *обрывки алгоритмов*, работающие с конечными аппроксимациями, способные воспринимать и запрашивать дополнительную входную информацию. Это в точности соответствует непрерывности как вычислимости.

Проблема

В системах доказательного программирования (AGDA) проверка равенства — базовая операция. Но:

Равенство действительных чисел $\Rightarrow$ проверка бесконечной информации

Не имеет вычислительного смысла!

Пути решения

1. **Заменить равенство приближенным:** Найти $x : |f(x)| < \varepsilon$, а не $f(x) = 0$
2. **Работать с классами эквивалентности:** Конкретное представление числа — "обрывок"
3. **Переопределить операции:** $x = y$ не влечет $x^2 = y^2$, а лишь $|x^2 - y^2| < \varepsilon$

В конструктивизме нет классической трихотомии

Вместо $x < y \lor x = y \lor x > y$ используется конструктивная аппроксимация:

$$\forall x, y\ \forall \varepsilon > 0((x > y - \varepsilon) \lor (x < y + \varepsilon))$$

Некорректный классический оператор

if (x < y) then S else T

Конструктивная (недетерминированная) замена

if (x < y + prec) then S else if (x > y - prec) then T

Или просто явный комментарий

if (x < y) then S else T // точность аргументов prec

Педагогическая проблема

Утверждение, которое вводит в заблуждение

“Рациональные числа — подмножество действительных” — **глобальная ошибка!**

Проблема

Избыточная информация (приближения) мешает работе с релевантной (точное отношение)

Правильный топологический подход

- Целые числа корректно представлять как частный случай действительных
- Рациональные числа как стабилизирующиеся последовательности

Цепные дроби

Идея для рациональных приближений

1. Сдерживать рост числителя (выделять целую часть)
2. Большой знаменатель $\Rightarrow$ выход за точность $\Rightarrow$ округление
3. **Оптимальный метод:** Разложить дробную часть в цепную дробь и отбросить последний член

Конечная информация о функции

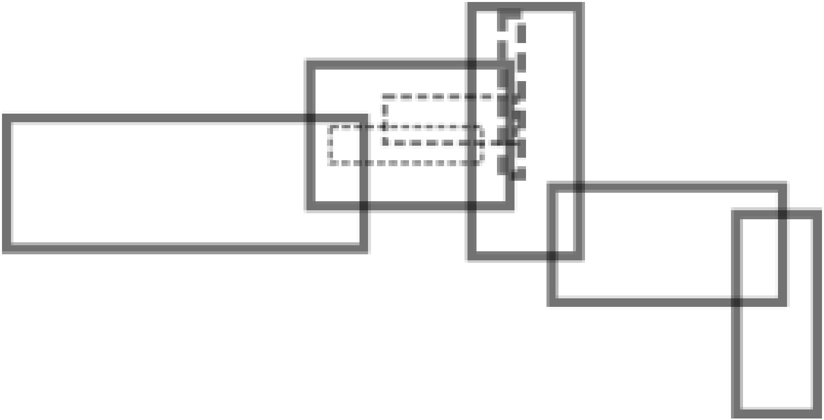

Рис.: Функция задана покрытием прямоугольниками

Комментарии

Свойства

- Пунктир — возможное уточнение информации
- **Уточнение** $b \preccurlyeq a$: каждый прямоугольник из b вложен в прямоугольник из a
- Позволяет приближённо вычислять интеграл

Важное наблюдение

Уточнение функции может **ухудшить** точность производной. Нет общего метода вычисления производной только по функции!

Конечная информация о функции с производной

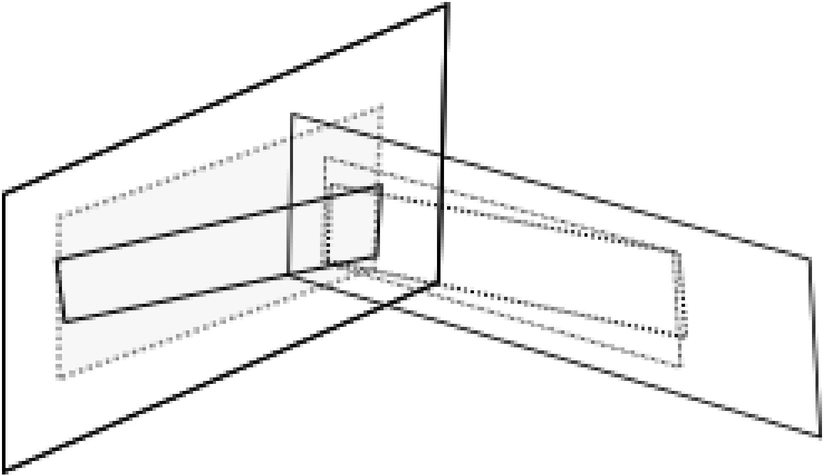

Рис.: Функция + производная заданы параллелограммами

Комментарии

Уточнение информации

Уточнение $b \preccurlyeq a$: параллелограмм из b вложен в параллелограмм из a **и** в его пропорциональную реплику

Связь координат $(x, y, z \approx f')$

$$y_2 - y_1 \leqslant |(x_2 - x_1) \times \max(|z_1|, |z_2|)|$$

Вывод

Для функций высших порядков нужна своя конечная информация

Большое конечное как непрерывное

Новый взгляд на большие данные

Большие данные (тексты, БД) можно рассматривать как бесконечные последовательности (бэровские пространства)

Конструктивный подход

- Функция работает с конечным фрагментом ("окрестностью")
- Если точности недостаточно — запрашивает уточнение данных
- **Модель:** интерактивные вычисления с внешним источником

Перспективы

Открывает путь для конструктивного анализа алгоритмов обработки больших данных

Конструктивизация спецификации

Следующий шаг в разработке надёжного ПО

Точность как часть спецификации

- Реальные данные всегда известны с некоторой точностью $(\pm\delta)$
- Требуемая точность результата $(\pm\varepsilon)$ — часть постановки задачи
- **Окрестностное отношение** — ключевая конструктивная характеристика

Как получить окрестностное отношение?

- **Легко:** комментируя программу при создании
- **Трудно:** анализируя готовый код
- **Практически невозможно:** тестируя "чёрный ящик"

Необходимость СПО

Для анализа кода и выявления таких отношений необходимы инструменты свободного ПО

Задачи для сообщества свободного ПО

Ближайшие цели

1. **Накопить библиотеку примеров** логически корректных непрерывных программ
2. **Выработать рекомендации и техники** для спецификации точности в коде
3. **Использовать практики СПО:**
   - Рецензирование кода сообществом
   - Системы отслеживания задач (Bugzilla) для фиксации требований к точности
   - Инструменты статического анализа

Цель

Повысить надёжность, предсказуемость и логическую корректность программ для работы с:

- Непрерывными данными
- Большими объектами

Спасибо за внимание!

Вопросы?

Дополнительные материалы

- Bridges D. et al. Handbook of Constructive Mathematics (2023)
- Непейвода Н. Н. Непрерывность как вычислимость (2025)
- Kreisel G., Troelstra A. S. Formal systems for some branches of intuitionistic analysis (1970)

Контактная информация:
Н. Н. Непейвода
ИПС РАН, Переславль-Залесский
email: nepejvodann@gmail.com