

Задачи к экзамену

Kakadu

16 декабря 2018 г.

1. Задача 1: мини-язык про трансформеры монад

Запишите AST для мини-языка и "интерпретатор" для него. Тип должен называться $M\ m\ r$, где

- M – это название вышележащей монады;
- по типу вложенной монады m язык будет полиморфен;
- r – типовый параметр вложенной монады. В некотором смысле, что что мы описываем должно быть изоморфно $M\ (m\ r)$.

```
class Transformer t where
  promote :: (Monad m) => m a -> t m a
  observe :: (Monad m) => t m a -> m a
```

Для нашей задачи создание терма мини-языка из вложенной монады будет называться **promote**, а вызов интерпретатора и возвращение ответа будет называться **observe**.

Для этой задачи будем считать, что вышележащей монадой является монада недетерминизма с бэктрекингом с операциями:

- Константа **false** которая символизирует, что ответ не нашли.
- Бинарная операция **|**, которая символизирует недетерминированный выбор.
- Операция **cons**, которая является полезным сокращением для **return a | m**

Что надо сделать (можно выполнять не обязательно в указанном порядке):

1. Опишите тип для мини-языка как алгебраический тип. Очевидно, там должны присутствовать конструкторы-аналоги для **return**, **>>=** и для **fail**. Плюс, конструкции специфичные для вышележащей монады (она нам известна). Никаких лямбд, арифметики и парсеров добавлять не нужно.
2. Напишите интерпретатор миниязыка (полиморфный по внутренней монаде).
3. Покажите, что то, что получилось – это монада (реализуйте три функции: **return**, **>>=** и **fail**). Естественным подходом будет использование композиции уже написанного интерпретатора и/или конструкторов, например **return = Return**

4. Проверьте, что законы монад выполняются.
5. Какие дополнительные, специфичные для вышележащей монады, законы разумно ввести?
6. Какова ассимптотическая сложность интерпретации в зависимости от длины термина мини-языка?
7. (Для оценки 5) Попробуйте каким-нибудь образом переписать интерпретатор так, чтобы можно было избавиться от необходимости введения конструкторов для описания структуры мини-языка (что по сути означает отсутствие необходимости задавать тип для мини-языка), а ограничиться только функциями и композицией функций. (Если вы переписали интерпретатор так, что справа от символа равенства не встречаются конструкторы – вы на пути к успеху). Делать это можно используя идеи `tagless final` стиля, или каким-то специфичным `ad hoc`’ом для данной задачи (мне всёравно).