

# Экстракция кода из Agda в Haskell

Шабалин Александр

научный руководитель  
доц. Москвин Д. Н.

Академический университет  
2014 г.

# Формальная верификация

- ▶ Необходимо уметь убеждаться, что написанная программа решает поставленную задачу.
- ▶ Тестирование не может показать, что программа верна для всех случаев (если, конечно, нельзя сделать полный перебор).
- ▶ Формальная верификация позволяет сравнить программу с формальной математической моделью и доказать их эквивалентность на всех входных данных.

- ▶ Один из способов формально верифицировать — строить формулы достаточно мощной логики над элементами программы и проверять их на этапе компиляции.
- ▶ Agda позволяет строить формулы на языке предикативной конструктивной логики. TODO: Definition of predicative logic?

# Использование верифицированного кода

Написание верифицированного алгоритма недостаточно — необходимо еще использовать этот код из «реальных» приложений. Подходы:

1. Использовать Agda для написания приложений целиком.
  - + Можно верифицировать больше кода.
  - Не Тьюринг-полный язык.
2. По коду на Agda генерировать код на другом языке
  - + Удобнее писать «реальный» код.
  - Необходимо поддерживать корректность кода при трансляции.

Второй пункт называется «экстракция программ» и используется в системе Coq.

# Цели и задачи

## Цель

По коду на Agda получить код на Haskell, которую можно использовать из программы на Haskell, не нарушая внутренние инварианты, поддерживаемые Agda.

Задачи: TODO

1. Придумать
2. Реализовать
3. Доказать

# Выводы