

# Программирование с зависимыми типами

## Зависимые типы и agda

Валерий Исаев

5 сентября 2016 г.

## Зависимые типы

Agda

# Мотивация

- ▶ Типизация в языках программирования позволяет выражать свойства программ.

# Мотивация

- ▶ Типизация в языках программирования позволяет выражать свойства программ.
- ▶ Более мощные системы типов позволяют выражать больше инвариантов программ.

# Мотивация

- ▶ Типизация в языках программирования позволяет выражать свойства программ.
- ▶ Более мощные системы типов позволяют выражать больше инвариантов программ.
- ▶ Зависимые типы являются венцом эволюции и позволяют полностью описывать спецификацию программы.

## Пример

- ▶ В просто типизированном языке, если мы хотим написать функцию сортировки, то мы можем приписать ей тип

$$\text{sort} : \text{List } a \rightarrow \text{List } a$$

## Пример

- ▶ В просто типизированном языке, если мы хотим написать функцию сортировки, то мы можем приписать ей тип

$$\text{sort} : \text{List } a \rightarrow \text{List } a$$

- ▶ Разумеется этот тип не описывает полностью спецификацию этой функции.

## Пример

- ▶ В просто типизированном языке, если мы хотим написать функцию сортировки, то мы можем приписать ей тип

$$\text{sort} : \text{List } a \rightarrow \text{List } a$$

- ▶ Разумеется этот тип не описывает полностью спецификацию этой функции.
- ▶ В языке с зависимыми типами мы можем задать ей следующий тип:

$$\text{sort} : \text{List } a \rightarrow \text{SortedList } a$$



## Пример

- ▶ В просто типизированном языке, если мы хотим написать функцию сортировки, то мы можем приписать ей тип

$$\text{sort} : \text{List } a \rightarrow \text{List } a$$

- ▶ Разумеется этот тип не описывает полностью спецификацию этой функции.
- ▶ В языке с зависимыми типами мы можем задать ей следующий тип:

$$\text{sort} : \text{List } a \rightarrow \text{SortedList } a$$

- ▶ Это всё еще не является полной спецификацией, но уже лучше.

## Альтернативы

- ▶ Если мы хотим описать тип сортированных списков, то нам нужно уметь выражать произвольные формулы.

## Альтернативы

- ▶ Если мы хотим описать тип отсортированных списков, то нам нужно уметь выражать произвольные формулы.
- ▶ Тогда мы можем определить этот тип следующим образом:

$$\{xs : \text{List } a \mid \forall i \leq j < \text{length}(xs), xs[i] \leq xs[j]\}$$

## Альтернативы

- ▶ Если мы хотим описать тип сортированных списков, то нам нужно уметь выражать произвольные формулы.
- ▶ Тогда мы можем определить этот тип следующим образом:

$$\{xs : \text{List } a \mid \forall i \leq j < \text{length}(xs), xs[i] \leq xs[j]\}$$

- ▶ Если мы хотим написать функцию `sort`, то нам нужно уметь не только выражать формулы в нашем языке, но и их доказательства.

# Альтернативы

- ▶ Если мы хотим описать тип сортированных списков, то нам нужно уметь выражать произвольные формулы.
- ▶ Тогда мы можем определить этот тип следующим образом:

$$\{xs : \text{List } a \mid \forall i \leq j < \text{length}(xs), xs[i] \leq xs[j]\}$$

- ▶ Если мы хотим написать функцию `sort`, то нам нужно уметь не только выражать формулы в нашем языке, но и их доказательства.
- ▶ То есть мы могли бы разделить язык на две части: отдельно программы и отдельно формулы и доказательства.

# Соответствие Карри-Говарда

- ▶ Зависимые типы предоставляют более гибкий и удобный подход.

# Соответствие Карри-Говарда

- ▶ Зависимые типы предоставляют более гибкий и удобный подход.
- ▶ Основная идея заключается в том, что формулы – это частный случай типов. А доказательство формулы – это программа соответствующего типа.

Формулы	$\perp$	$\top$	$\rightarrow$	$\wedge$	$\vee$
Типы	Empty	Unit	$\rightarrow$	$\times$	Either

# Соответствие Карри-Говарда

- ▶ Зависимые типы предоставляют более гибкий и удобный подход.
- ▶ Основная идея заключается в том, что формулы – это частный случай типов. А доказательство формулы – это программа соответствующего типа.

Формулы	$\perp$	$\top$	$\rightarrow$	$\wedge$	$\vee$
Типы	Empty	Unit	$\rightarrow$	$\times$	Either

- ▶ Так как формулы и доказательства являются частным случаем типов и программ, нет необходимости иметь два отдельных языка.



# Зависимые типы

- ▶ При помощи простых типов можно выражать только формулы пропозициональной логики.

# Зависимые типы

- ▶ При помощи простых типов можно выражать только формулы пропозициональной логики.
- ▶ Если мы хотим формулировать интересные утверждения, нам также нужны кванторы.

# Зависимые типы

- ▶ При помощи простых типов можно выражать только формулы пропозициональной логики.
- ▶ Если мы хотим формулировать интересные утверждения, нам также нужны кванторы.
- ▶ Аналогами кванторов являются зависимые типы.

# Зависимые типы

- ▶ При помощи простых типов можно выражать только формулы пропозициональной логики.
- ▶ Если мы хотим формулировать интересные утверждения, нам также нужны кванторы.
- ▶ Аналогами кванторов являются зависимые типы.
- ▶ Зависимое произведение  $\prod (x : A) B$  является аналогом квантора всеобщности  $\forall (x : A) B$  и обобщает тип функций  $A \rightarrow B$ ,

# Зависимые типы

- ▶ При помощи простых типов можно выражать только формулы пропозициональной логики.
- ▶ Если мы хотим формулировать интересные утверждения, нам также нужны кванторы.
- ▶ Аналогами кванторов являются зависимые типы.
- ▶ Зависимое произведение  $\prod(x : A)B$  является аналогом квантора всеобщности  $\forall(x : A)B$  и обобщает тип функций  $A \rightarrow B$ ,
- ▶ Зависимая сумма  $\sum(x : A)B$  является аналогом квантора существования  $\exists(x : A)B$  и обобщает тип произведений  $A \times B$ .

# Применения

- ▶ Языки с зависимыми типами используют для двух различных целей.

# Применения

- ▶ Языки с зависимыми типами используют для двух различных целей.
- ▶ Во-первых, для верификации программ.

# Применения

- ▶ Языки с зависимыми типами используют для двух различных целей.
- ▶ Во-первых, для верификации программ.
- ▶ Во-вторых, так как такой язык является полноценной логикой, то его можно использовать для формализации математики.



# Реализации

- ▶ Существует несколько языков с зависимыми типами: `agda`, `coq`, `nuprl`, `idris`, ...

# Реализации

- ▶ Существует несколько языков с зависимыми типами: `agda`, `coq`, `nuprl`, `idris`, ...
- ▶ Мы будем использовать `agda`.