Язык Agda.

Язык Agda — это написанный на Haskell язык с зависимыми типами и средство для интерактивного доказательства теорем.
Рекомендации по установке:
I. MacOS (выполнить следующую последовательность скриптов в терминале):
1) Установить пакетный менеджер Homebrew (https://brew.sh/);
2) Установить Agda через Homebrew: `brew install agda`;
3) Установить текстовый редактор Emacs: `brew install emacs -with-cocoa`;

II. Linux:
1) Выполнить следующий скрипт: `sudo apt-get install agda-mode`;
2) Установить стандартную библиотеку: `apt-get install agda-stdlib`;

Работа с Agda как правило производится в текстовом редакторе Emacs, как правило скрипты для Emacs добавляются автоматически в нужное место при установке Agda.

Базовые определения HoTT в Agda.

1) Пути:
$data\ Id\ \{A : Type\}(a : A) : A \to Type$ where
$id : Id\ A\ A$

$Path : \{A : Type\} \to A \to A \to Type$
$Path = id$

Базовые определения HoTT в Agda.

2) Правило подстановки для путей:

$transport : \{B : Type\}(E : B \to Type)\{b_1\ b_2 : B\} \to Path\ b_1 b_2 \to E\ b_1 \to E\ b_2$

$transport\ E\ id = \lambda x \to x$

3) Факт: отображения между типами сохраняют пути:

$ap : \{A\ B : Type\}\{a_1\ a_2 : A\}(f : A \to B) \to Path\ a_1\ a_2 \to Path\ (fa_1)\ (fa_2);$

$ap\ f\ id = id.$

4) Элиминатор для путей:

$coe : \{A\ B : Type\} \to Path\ A\ B \to A \to B$

$coe = transport\ (\lambda x \to x)$

Базовые определения HoTT в Agda.

5. Объявление для сигма-типа, типа зависимой пары:

$record\ \Sigma\ (A : Type)(B : A \to Type) : Type$ where

$\pi_1 : A$

$\pi_2 : B\ \pi_1$

Базовые определения HoTT в Agda.

6. Эквивалентность типов:
$\mathit{record}\ \mathit{IsEquiv}\{AB : \mathit{Type}\}(f : A \to B)$ where
$g : B \to A$
$\alpha : (x : A) \to \mathit{Path}\,(f\ (g\ x))\ x$
$\beta : (y : B) \to \mathit{Path}\,(g\ (f\ y))\ y$
$\gamma : (x : A) \to \mathit{Path}\,(\beta\ (f\ x))(\mathit{ap}\ f\ (\alpha\ x))$

Эквивалетность между типами можно отдельной фукцией, в сигнатуре которой мы перечисляем все методы в качестве аргументов, а возвращаем терм типа $\mathit{Equiv}\ A\ B$.

Слабая эквивалетность задается аналогично, но без метода γ.

Базовые определения HoTT в Agda.

7. Унивалетность устанавливает эквивалентность между эквивалетностью между A и B, с одной стороны, и путями A и B с другой. Эквивалентная формулировка существует для эквивалентности и слабой эквивалентности.
postulate
$univalence : \{A\ B : Type\} \rightarrow IsEquiv\{Path\ A\ B\}\{EquivA\ B\}$

$ua : \{A\ B : Type\} \rightarrow Equiv\ A\ B \rightarrow Path\ A\ B$

Вычисление гомотопической группы окружности.

(Дисклеймер: функции имеют слишком длинную реализацию, поэтому мы будем в таком случае указывать одни лишь их типы).

8. Определение типа окружности:

$S^1 : Set$

$base : S^1$

$loop : base \equiv base$

9. Индукция и рекурсия для S^1:

$S^1 - rec : \{A : Type\} \to (a : A) \to (\alpha : a \equiv a) \to S^1 \to A$

$Cover : S^1 \to Type$

$Cover = S^1 - rec\ \mathbb{Z}\ (ua\ succEquiv)\ x$

Вычисление гомотопической группы окружности.

10. Кодирование:
$encode : \{x : S^1\} \to Path\ base\ x \to Cover\ x$
$encode\ \alpha = transport\ Cover\ \alpha\ Zero$

11. Накручивание петли:
$loop^\wedge : \mathbb{Z} \to Path\ base\ base$
$loop^\wedge\ Zero = id$
$loop^\wedge\ (Pos\ One) = loop$
$loop^\wedge\ (Pos\ (Succ\ n)) = loop \circ loop^\wedge\ n$
$loop^\wedge\ (Neg\ One) =!\ loop$
$loop^\wedge\ (Neg(Succ\ n)) =!\ loop \circ loop^\wedge(Neg\ n)$.

12. Накручивание сохраняет функцию следования (аналогично для предшествования):
$loop^\wedge\ succ : (n : \mathbb{Z}) \to Path(loop \circ loop^\wedge\ n)(loop^\wedge(Succ\ n))$

Вычисление гомотопической группы окружности.

13. Свойство кодирования петли:
$encode - loop^{\wedge} : (n : \mathbb{Z}) \to Cover \to Path\,(encode\,(loop^{\wedge}\,n))\ n$

14. Декодирование: $decode : \{x : S^1\} \to Cover\ x \to Path\ base\ x$.

15. Кодирование и декодирование взаимнообратны:
$encode - decode : \{x : S^1\} \to (c : Cover\ x) \to Path(encode\,(decode\,\{x\}\ c))\ c$
$decode - encode : \{x : S^1\}(\alpha : Path\ base\ x) \to Path(decode(encode\ \alpha))\ \alpha$

16. Слабая эквивалентность между S^1 и $\mathbb{Z}$.
$\Omega S^1 \cong \mathbb{Z} : HEquiv(Path\ base\ base)\ \mathbb{Z}$

Используя слабую эквивалентность, можно получить просто эквивалетность, используя унивалентность