Homotopy Type Theory

in Agda

17/7/7

Goal

synthetic homotopy theory in Agda

+ other needed theories

Goal

synthetic homotopy theory in Agda

+ other needed theories

Agda and Coq were the only two
immediately usable systems for HoTT

Decentralized Dev.

`HoTT/Agda-HoTT`

`favonia/homotopy` [obsolete]

`nicolaikraus/HoTT-Agda` [fork]

`dlicata335/hott-agda`

`guillaumebrunerie/JamesConstruction`

...

Decentralized Dev.

`HoTT/Agda-HoTT`

`favonia/homotopy` [obsolete]

`nicolaikraus/HoTT-Agda` [fork]

`dlicata335/hott-agda`

`guillaumebrunerie/JamesConstruction`

. . .

porting theorems and forking are common

Decentralized Dev.

`HoTT/Agda-HoTT`

`favonia/homotopy [obsolete]`

`nicolaikraus/HoTT-Agda [fork]`

`dlicata335/hott-agda`

`guillaumebrunerie/JamesConstruction`

. . .

porting theorems and forking are common

HoTT/Agda-HoTT

- generalized Blakers-Massey (WIP)
- total space of Hopf, 3x3 lemma
- Seifert-van Kampen theorem
- Mayer–Vietoris sequences
- cubical reasoning
- Freudenthal suspension theorem
- Eilenberg-MacLane spaces K(G,n)
- ...

Guillaume Brunerie, Kuen-Bang Hou (Favonia),
Evan Cavallo, Eric Finster, Jesper Cockx,
Christian Sattler, Chris Jeris and Michael Shulman

Used Features

- MLTT-style logic/programming languages
- inductive-inductive & inductive-recursive
- powerful mixfix parser
- pattern matching
- universe polymorphism
- ...

Used Features

- MLTT-style logic/programming languages
- inductive-inductive & inductive-recursive
- powerful mixfix parser
- pattern matching
- universe polymorphism
- ...

Used Automation

- higher-order unification
- literal overloading
- FEW tactics

Higher Inductive Types?

Simulated by **rewriting rules** in HoTT-Agda

Higher Inductive Types?

Simulated by **rewriting rules** in HoTT-Agda

```
postulate
  S¹ : Type₀
  base : S¹
  loop : base == base

module S¹Elim {l}{P : S¹ → Type l}
 (base* : P base) (loop* : base* == base* [ P ↓ loop ]) where

 postulate
    f : Π S¹ P
    base-β : f base ↦ base*
 {-# REWRITE base-β #-}

 postulate
    loop-β : apd f loop == loop*
```

* effectively the same as Dan's trick

Semantics of Agda

- NOT well-understood (as a whole)
- Many individual features proved

Semantics of Agda

- NOT well-understood (as a whole)
- Many individual features proved

Mode of Usage

- Highly experimental

Structures and Stats

`core/` [10520 code + 1024 comments]

basic synthetic homotopy theory

`theorems/` [16107 code + 1577 comments]

interesting results

continuous integration through travis

the entire codebase can be checked in 20-30 mins