An Introduction to Homotopy Type Theory

Amr Sabry

School of Informatics and Computing
Indiana University

October 31, 2013

Higher-Order Inductive Types

- We cannot generate non-trivial groupoids starting from the usual type constructions;
- We need higher-order inductive types
- A new development (2012)
- You specify not only the points in the “set” but also the various (iterated) paths

Higher-Order Inductive Types (example)

```
- data Circle :  Set where
- base :  Circle
- loop :  base ≡ base
```

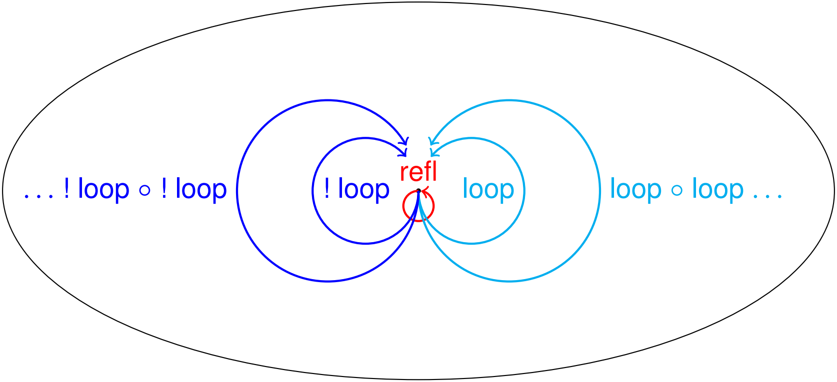
module Circle where
  private data S^{1*} : Set where base* : S^{1*}

  S^1 : Set
  $S^1 = S^{1*}$

  base : S^1
  base = base*

  postulate loop : base ≡ base

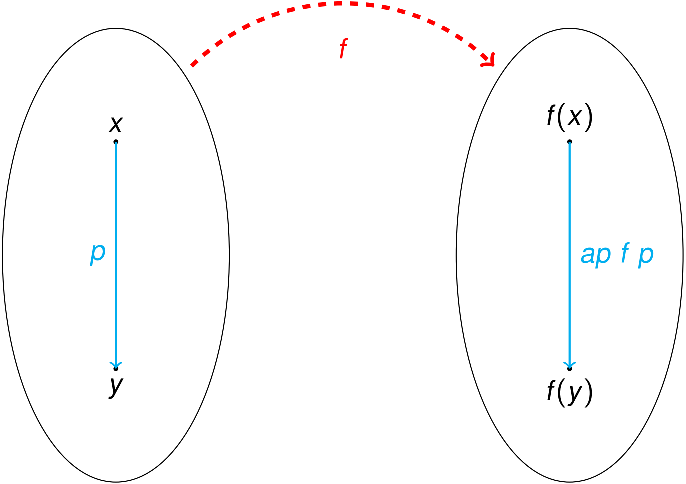
Non-trivial structure of this example



Functions as functors

- A function from space A to space B must map the points of A to the points of B as usual but it must also respect the path structure
- Mathematically, this corresponds to saying that every function respects equality;
- Topologically, this corresponds to saying that every function is continuous.

Functions as functors



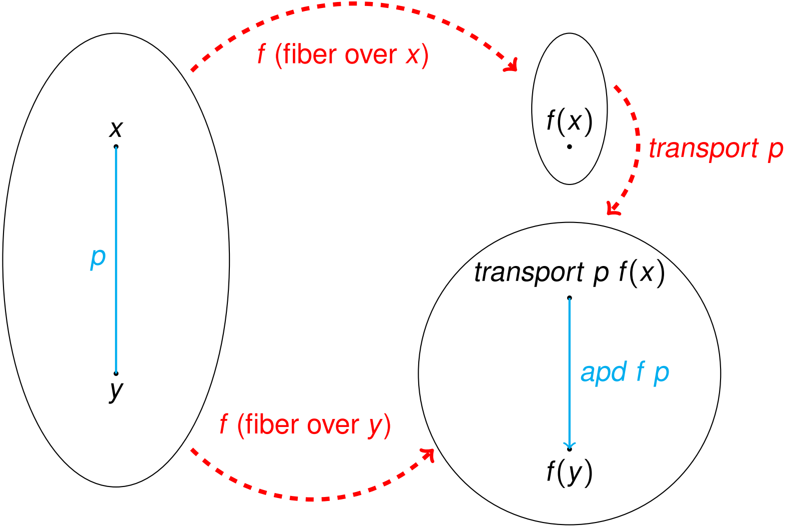
Functions as functors

- *ap f p* is the action of *f* on a path *p*;
- This satisfies the following properties:
  - $ap\ f\ (p \circ q) \equiv (ap\ f\ p) \circ (ap\ f\ q)$;
  - $ap\ f\ (!\ p) \equiv\ !\ (ap\ f\ p)$;
  - $ap\ g\ (ap\ f\ p) \equiv ap\ (g \circ f)\ p$;
  - $ap\ id\ p \equiv p$.

Type families as fibrations

- A more complicated version of the previous idea for dependent functions;
- The problem is that for dependent functions $f(x)$ and $f(y)$ may not be in the same type, i.e., they live in different spaces;
- Idea is to transport $f(x)$ to the space of $f(y)$;
- Because everything is "continuous", the path p induces a transport function that does the right thing.

Type families as fibrations



Extensional Equivalences

Univalence