The Flattening Lemma

(Seminar of Homotopy Type Theory)

Jonathan Prieto-Cubides

Universitetet i Bergen
Bergen, Norway

May 28, 2018

Updated: July 25, 2018

Overview

- Motivation
- The Circle
- Univalence and Transport
- The Flattening Lemma (FL)
- Example using FL
- Alternative Lemma to FL
- Proof of FL

Motivation:

We want to prove

$$\left(\sum_{x:\mathbb{S}^1} B(x)\right) \simeq \mathbb{S}^1$$

where the **univalence axiom** is used to define the type family $B : \mathbb{S}^1 \to \mathcal{U}$ as follows.

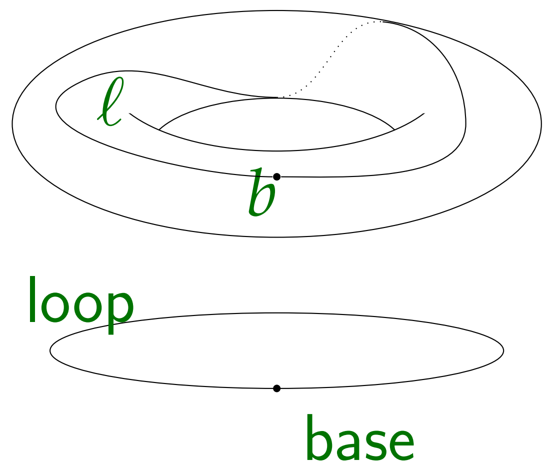
$$B(\mathsf{base}) :\equiv \mathbf{2} \qquad \text{and} \qquad B(\mathsf{loop}) := \mathsf{ua}(\mathsf{neg}).$$

(Whiteboard. Def. of the equivalence)

The Circle ($\mathbb{S}^1$)

The higher inductive type $\mathbb{S}^1$ is generated by

- A point $\mathsf{base} : \mathbb{S}^1$, and
- A path $\mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}$.



Lemma 1.1.1. *If A is a type together with $a : A$ and $p : a =_A a$, then there is a function $f : \mathbb{S}^1 \to A$ with*

$$f(\mathsf{base}) :\equiv a,$$
$$\mathsf{ap}_f(\mathsf{loop}) := p.$$

The induction principle for $\mathbb{S}^1$:

- given $P : \mathbb{S}^1 \to \mathcal{U}$,
- an element $b : P(\mathsf{base})$, and
- a path $\ell : b =^P_{\mathsf{loop}} b$,

$$\mathsf{ind}_{\mathbb{S}^1} : \sum_{b:P(\mathsf{base})} \mathsf{transport}^P(\mathsf{loop}, b) = b \to \prod_{x:\mathbb{S}^1} P(x).$$

There exists a function $f : \prod_{(x:\mathbb{S}^1)} P(x)$ such that

$$\begin{aligned} f &:\equiv \mathsf{ind}_{\mathbb{S}^1}(b, \ell), \\ f(\mathsf{base}) &\equiv b, \\ \mathsf{apd}_f(\mathsf{loop}) &= \ell. \end{aligned}$$

Univalence and Transport

- An introduction rule for $(A =_{\mathcal{U}} B)$:

$$\mathsf{ua} : (A \simeq B) \to (A =_{\mathcal{U}} B).$$

- The elimination rule for $(A =_{\mathcal{U}} B)$:

$$\mathsf{idtoeqv} \equiv \mathsf{transport}^{X \mapsto X} : (A =_{\mathcal{U}} B) \to (A \simeq B).$$

Lemma 1.1.2. *For any type family $B : A \to \mathcal{U}$ and $x, y : A$ with a path $p : x = y$ and $u : B(x)$, we have*

$$\begin{aligned}\mathsf{transport}^{B}(p, u) &= \mathsf{transport}^{X \mapsto X}(\mathsf{ap}_B(p), u)\\ &= \mathsf{idtoeqv}(\mathsf{ap}_B(p))(u).\end{aligned}$$

Lemma 1.1.3. *Given $B : A \to \mathcal{U}$ and $x, y : A$, with a path $p : x = y$ and an equivalence $e : B(x) \simeq B(y)$ such that $\mathsf{ap}_B(p) = \mathsf{ua}(e)$, then for any $u : B(x)$ we have*

$$\mathsf{transport}^{B}(p, u) = e(u).$$

(Pictures on the whiteboard)

Suppose we have a type X and the equivalence $e : X \simeq X$. We can define a type family $B : \mathbb{S}^1 \to \mathcal{U}$ by using $\mathbb{S}^1$-recursion:

$$B(\mathsf{base}) :\equiv X \qquad \text{and} \qquad B(\mathsf{loop}) := \mathsf{ua}(e).$$

The type X thus appears as the fiber $B(\mathsf{base})$ of B over the basepoint. We consider as the equivalence e the boolean negation ($\mathsf{neg} : \mathbf{2} \simeq \mathbf{2}$). Therefore,

$$\begin{aligned} B &:\equiv \mathsf{rec}_{\mathbb{S}^1}(\mathcal{U}, \mathbf{2}, \mathsf{ua}(\mathsf{neg})), \\ B(\mathsf{base}) &:\equiv \mathbf{2}, \\ B(\mathsf{loop}) &:= \mathsf{ua}(\mathsf{neg}). \end{aligned}$$

(Pictures on the whiteboard.)

If $x : \mathbf{2}$ and B as was defined above:

Lemma 1.1.4. $\mathsf{transport}^B(\mathsf{loop}, x) = \mathsf{neg}(x)$.

Proof. **??** $(B, \mathsf{loop}, \mathsf{neg}, x)$. □

Lemma 1.1.5. *Suppose $A : \mathcal{U}$, that $x, y, z, w : A$ and that $p : x = y$ and $q : y = z$ and $r : z = w$. We have the following:*

(i). $p^{-1} \cdot p = \mathsf{refl}_y$ *and* $p \cdot p^{-1} = \mathsf{refl}_x$.

Lemma 1.1.6. *Given $P : A \to \mathcal{U}$ with $p : x =_A y$ and $q : y =_A z$ while $u : P(x)$, we have*

$$q_*(p_*(u)) = (p \cdot q)_*(u).$$

Lemma 1.1.7. $\mathsf{transport}^B(\mathsf{loop}^{-1}, x) = \mathsf{neg}(x)$.

Lemma 1.1.8. $\mathsf{transport}^B(\mathsf{loop}^2, x) = x$.

Whiteboard.

Lemma 1.1.9 (Dependent map). *Suppose $f : \prod_{(x:A)} P(x)$; then we have a map*

$$\mathsf{apd}_f : \prod_{p:x=y} \big(p_*(f(x)) =_{P(y)} f(y)\big).$$

Lemma 1.1.10. *If $P : A \to \mathcal{U}$ is defined by $P(x) :\equiv B$ for a fixed $B : \mathcal{U}$, then for any $x, y : A$ and $p : x = y$ and $b : B$ we have a path*

$$\mathsf{transportconst}^B_p(b) : \mathsf{transport}^P(p, b) = b.$$

Lemma 1.1.11. *For $f : A \to B$ and $p : x =_A y$, we have*

$$\mathsf{apd}_f(p) = \mathsf{transportconst}^B_p(f(x)) \cdot \mathsf{ap}_f(p).$$

Theorem 1.1.12. *Suppose that $P : A \to \mathcal{U}$ is a type family over a type A and let $w, w' : \sum_{(x:A)} P(x)$. Then there is an equivalence*

$$(w = w') \simeq \sum_{(p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w'))} p_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w').$$

Lemma 1.1.13. *Given a type X, a path $p : x_1 =_X x_2$, type families $A, B : X \to \mathcal{U}$, and a function $f : A(x_1) \to B(x_1)$, we have*

$$\mathsf{transport}^{x \mapsto A(x) \to B(x)}(p, f) =$$

$$\left(x \mapsto \mathsf{transport}^{B}(p, f(\mathsf{transport}^{A}(p^{-1}, x)))\right)$$

Theorem 1.1.14. *For $f, g : A \to B$, with $p : a =_A a'$ and $q : f(a) =_B g(a)$, we have*

$$\mathsf{transport}^{x \mapsto f(x) =_B g(x)}(p, q) =_{f(a') = g(a')} (\mathsf{ap}_f p)^{-1} \cdot q \cdot \mathsf{ap}_g p.$$

Theorem 1.1.15. *Let $B : A \to \mathcal{U}$ and $f, g : \prod_{(x:A)} B(x)$, with $p : a =_A a'$ and $q : f(a) =_{B(a)} g(a)$. Then we have*

$$\mathsf{transport}^{x \mapsto f(x) =_{B(x)} g(x)}(p, q) = (\mathsf{apd}_f(p))^{-1} \cdot \mathsf{ap}_{(\mathsf{transport}^B p)}(q) \cdot \mathsf{apd}_g(p).$$

The Flattening Lemma[1]

- If W and $\widetilde{W}$ are two higher inductive types,
- $P : W \to \mathcal{U}$ is a type family over W, and
- $\widetilde{W}$ constructors can be deduced from W constructors and from the definition of P

We have the equivalence

$$\left(\sum_{x:W} P(x)\right) \simeq \widetilde{W}.$$

$\widetilde{W}$ is called the "flattened" higher inductive of the total space $\sum_{(x:W)} P(x)$.

[1]See the full description in **??**.

Lemma 1.1.16 (The Flattening Lemma).

$$\left(\sum_{x:W} P(x)\right) \simeq \widetilde{W}$$

Suppose we have $A, B : \mathcal{U}$ *and* $f, g : B \to A$, *and that the higher inductive type* W *is generated by*

- $\mathsf{c} : A \to W$
- $\mathsf{p} : \prod_{(b:B)} (\mathsf{c}(f(b)) =_W \mathsf{c}(g(b)))$

In addition, suppose we have

- $C : A \to \mathcal{U}$
- $D : \prod_{(b:B)} C(f(b)) \simeq C(g(b))$

Define a type family $P : W \to \mathcal{U}$ *recursively by*

$$P(\mathsf{c}(a)) :\equiv C(a) \qquad \text{and} \qquad P(\mathsf{p}(b)) := \mathsf{ua}(D(b)).$$

Let $\widetilde{W}$ *be the higher inductive type generated by*

- $\widetilde{\mathsf{c}} : \prod_{(a:A)} C(a) \to \widetilde{W}$ *and*
- $\widetilde{\mathsf{p}} : \prod_{(b:B)} \prod_{(y:C(f(b)))} (\widetilde{\mathsf{c}}(f(b), y) =_{\widetilde{W}} \widetilde{\mathsf{c}}(g(b), D(b)(y)))$.

Exercise:

$$\sum_{x:\mathbb{S}^1} B(x) \simeq \mathbb{S}^1.$$

Solution.
We show first the equivalence,

$$\sum_{x:\mathbb{S}^1} B(x) \simeq \Sigma\mathbf{2},$$

by applying **??** with the definitions:

- $W :\equiv \mathbb{S}^1$,
- $A :\equiv B :\equiv \mathbf{1}$, $f :\equiv g :\equiv id_{\mathbf{1}}$.

- $\mathsf{c} : \mathbf{1} \to W$ with $\mathsf{c} :\equiv \lambda\,\_\,.\,\mathsf{base}$,
- $\mathsf{p} : \prod_{(b:\mathbf{1})} \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}$ with $\mathsf{p} :\equiv \lambda\,\_\,.\,\mathsf{loop}$.

- $C : \mathbf{1} \to \mathcal{U}$ with $C :\equiv \lambda\,\_\,.\,\mathbf{2}$,
- $D : \prod_{(b:B)} \mathbf{2} \simeq \mathbf{2}$ with $D :\equiv \lambda\,\_\,.\,\mathsf{neg}$.

- The type family $P : \mathbb{S}^1 \to \mathcal{U}$, $P :\equiv B$.

$$B(\mathsf{base}) :\equiv \mathbf{2}, \;\; \mathsf{ap}_B(\mathsf{loop}) :\equiv \mathsf{ua}(\mathsf{neg}).$$

- $\widetilde{W} :\equiv \Sigma\mathbf{2}$.

- $\widetilde{\mathsf{c}} : \prod_{(a:\mathbf{1})} \mathbf{2} \to \Sigma\mathbf{2}$,
  $\widetilde{\mathsf{c}} :\equiv \lambda\_.\mathsf{rec}_{\mathbf{2}}(\Sigma\mathbf{2}, \mathsf{N}, \mathsf{S})$.

- $\widetilde{\mathsf{p}} : \prod_{(b:\mathbf{1})} \prod_{(y:\mathbf{2})} \widetilde{\mathsf{c}}(b, y) =_{\widetilde{W}} \widetilde{\mathsf{c}}(b, \mathsf{neg}(y))$,
  $\widetilde{\mathsf{p}} :\equiv \lambda\_.\mathsf{ind}_{(y:\mathbf{2})}(\mathsf{merid}(0), \mathsf{merid}(1)^{-1})$.

In class, we proved that $\Sigma\mathbf{2} \;\simeq\; \mathbb{S}^1$. Then, by transitivity of our equivalence relation ($\simeq$), we have

$$\sum_{x:\mathbb{S}^1} Bx \;\simeq\; \mathbb{S}^1.$$

$\square$

Alternative Approach:[2]

Lemma 1.1.17. *For* $A : \mathcal{U}$, $B : A \to \mathcal{U}$, $C : \mathcal{U}$,

$$\sum_{a:A} B(a) \simeq C,$$

if we have

- $f : \prod_{(a:A)} (B(a) \to C)$,
- $g : C \to A$,
- $h : \prod_{(c:C)} B(g(c))$,
- $\boldsymbol{\alpha : \prod_{(c:C)} f(g(c), h(c)) =_C c}$,
- $\boldsymbol{\beta_0 : \prod_{(a:A)} \prod_{(b:B(a))} g(f(a,b)) =_A a}$, *and*
- $\boldsymbol{\beta_1 : \prod_{(a:A)} \prod_{(b:B(a))} \mathsf{transport}^B(\beta_0(a,b), h(f(a,b))) = b}$.

[2] Suggested and verified in AGDA by Håkon Robbestad

Proof. Let us define

- $\hat{f} : \sum_{(a:A)} B(a) \to C$
  $\hat{f}((a,b)) :\equiv f(a,b)$,
- $\hat{f}^{-1} : C \to \sum_{(a:A)} B(a)$,
  $\hat{f}^{-1}(b) :\equiv (g(c), h(c))$.
- $s_1 : \prod_{(c:C)} \hat{f}(\hat{f}^{-1}(c)) = c$,
  $s_1 :\equiv \alpha$.
- $s_2 : \prod_{((a,b):\sum_{(x:A)} B(x))} \hat{f}^{-1}(\hat{f}(a,b)) = (a,b)$,
  $s_2 :\equiv \lambda(a,b).\mathbf{??}^{-1}(\beta_0(a,b), \beta_1(a,b))$.

Then,

$$(\hat{f}, ((\hat{f}^{-1}, s_1), (\hat{f}^{-1}, s_2))) : \sum_{a:A} B(a) \ \simeq \ C.$$

$\square$

Proof of ??. (Section 6.12 Pp. 273-289).

Lemma 1.1.18. *There are functions*

- $\widetilde{\mathsf{c}}' : \prod_{(a:A)} C(a) \to \sum_{(x:W)} P(x)$ *and*
- $\widetilde{\mathsf{p}}' : \prod_{(b:B)} \prod_{(y:C(f(b)))}$
  $$\left(\widetilde{\mathsf{c}}'(f(b), y) =_{\sum_{(w:W)} P(w)} \widetilde{\mathsf{c}}'(g(b), D(b)(y))\right).$$

Proof.

- Define $\widetilde{\mathsf{c}}'(a, x) :\equiv (\mathsf{c}(a), x)$.
- Given $b : B$ and $y : C(f(b))$, since we have
  $$\mathsf{p}(b) : \mathsf{c}(f(b)) = \mathsf{c}(g(b)),$$
  $$\widetilde{\mathsf{p}}'(b, y) : (\mathsf{c}(f(b)), y) = (\mathsf{c}(g(b)), D(b)(y))$$
  $$\widetilde{\mathsf{p}}'(b, y) :\equiv \lambda by.\mathbf{??}^{-1}$$
  $$(\mathsf{p}(b), \mathbf{??}(P, D, \mathsf{p}(b), y)).$$

□

Lemma 1.1.19. *Suppose* $Q : \left(\sum_{(x:W)} P(x)\right) \to \mathcal{U}$ *is a type family and that we have*

- $c : \prod_{(a:A)} \prod_{(x:C(a))} Q(\widetilde{\mathsf{c}}'(a,x))$ *and*
- $p : \prod_{(b:B)} \prod_{(y:C(f(b)))}$
$$\left(\widetilde{\mathsf{p}}'(b,y)_*(c(f(b),y)) = c(g(b),D(b)(y))\right).$$

Then there exists $k : \prod_{(z:\sum_{(w:W)} P(w))} Q(z)$ *such that* $k(\widetilde{\mathsf{c}}'(a,x)) \equiv c(a,x)$.

Proof. Proof in the book, Pp. 276-277. □

Lemma 1.1.20. *Suppose* Q *is a type and that we have*

- $c : \prod_{(a:A)} C(a) \to Q$ *and*
- $p : \prod_{(b:B)} \prod_{(y:C(f(b)))} \Big(c(f(b), y) =_Q c(g(b), D(b)(y))\Big)$.

Then there exists $k : \left(\sum_{(w:W)} P(w)\right) \to Q$ *such that* $k(\widetilde{\mathsf{c}}'(a, x)) \equiv c(a, x)$.

Proof. Proof in the book, Pp. 277-278. □

Lemma 1.1.21. *Let* $B : X \to \mathcal{U}$ *be a type family and let*

$$f : \sum_{x:X} B(x) \to Z$$

be defined componentwise by $f(x, b) :\equiv d(x)(b)$ *for a curried function* $d : \prod_{(x:X)} B(x) \to Z$. *Then if we have*

- $s : x_1 =_X x_2$,
- $b_1 : B(x_1), b_2 : B(x_2)$, *and*
- $r : s_*(b_1) = b_2$,

the path

$$\mathsf{ap}_f(\mathsf{pair}^=(s, r)) : f(x_1, b_1) = f(x_2, b_2)$$

is equal to the composite . . .

the path

$$\mathsf{ap}_f(\mathsf{pair}^{=}(s,r)) : f(x_1,b_1) = f(x_2,b_2)$$

is equal to the composite ...

$$\begin{aligned}
f(x_1,b_1) &\equiv d(x_1)(b_1) \\
&= \mathsf{transport}^{x\mapsto Z}(s, d(x_1)(b_1)) && \text{(by } (\mathbf{??})^{-1}\text{)} \\
&= \mathsf{transport}^{x\mapsto Z}(s, d(x_1)((\mathsf{refl}_{x_2})_*(b_1))) && \text{(by } (\mathbf{??})^{-1}\text{)} \\
&= \mathsf{transport}^{x\mapsto Z}(s, d(x_1)((s^{-1} \cdot s)_*(b_1))) && \text{(by } (\mathbf{????})^{-1}\text{)} \\
&= \mathsf{transport}^{x\mapsto Z}(s, d(x_1)(s^{-1}{}_*(s_*(b_1)))) && \text{(by } (\mathbf{??})^{-1}\text{)} \\
&= \big(\mathsf{transport}^{x\mapsto (B(x)\to Z)}(s, d(x_1))\big)(s_*(b_1)) && \text{(by (2))} \\
&= d(x_2)(s_*(b_1)) && \text{(by } \mathsf{happly}(\mathsf{apd}_d(s))(s_*(b_1))\text{)} \\
&= d(x_2)(b_2) && \text{(by } \mathsf{ap}_{d(x_2)}(r)\text{)} \\
&\equiv f(x_2,b_2).
\end{aligned}$$

Lemma 1.1.22. $\mathsf{ap}_f(\widetilde{\mathsf{p}}'(b,y)) = p(b,y)$.

Proof of **??** *(The Flattening Lemma).*
Let us define

- $k : (\sum_{(w:W)} P(w)) \to \widetilde{W}$ by using the recursion principle of **??**, with $\widetilde{\mathsf{c}}$ and $\widetilde{\mathsf{p}}$ as input data.
- $h : \widetilde{W} \to \sum_{(w:W)} P(w)$ by using the recursion principle for $\widetilde{W}$, with $\widetilde{\mathsf{c}}'$ and $\widetilde{\mathsf{p}}'$ as input data.

- $s_1 : \prod_{(z:\widetilde{W})} k(h(z)) = z$
  By induction on z, it suffices to consider the two constructors of $\widetilde{W}$. But we have
  $$k(h(\widetilde{\mathsf{c}}(a,x))) \equiv k(\widetilde{\mathsf{c}}'(a,x)) \equiv \widetilde{\mathsf{c}}(a,x)$$
  by definition, while similarly
  $$k(h(\widetilde{\mathsf{p}}(b,y))) = k\big(\widetilde{\mathsf{p}}'(b,y)\big) = \widetilde{\mathsf{p}}(b,y)$$
  using the propositional computation rule for $\widetilde{W}$ and **??**.
- $s_2 : \prod_{(z:\sum_{(w:W)} P(w))} h(k(z)) = z.$
  But this is essentially identical, using **??** for "induction on $\sum_{(w:W)} P(w)$" and the same computation rules.

$\square$

References

- Univalent Foundations Program, The (2013). Homotopy Type Theory: Univalent Foundations of Mathematics. Institute for Advanced Study. URL: http://saunders.phil.cmu.edu/book/hott-online.pdf
- Bezem M. Lecture notes for a seminar on Homotopy Type Theory. URL: https://github.com/marcbezem/INF329

(Bonus Slides)

Dependent pair types (Σ-types)

- if $A : \mathcal{U}_n$ and $B : A \to \mathcal{U}_n$, then $\sum_{(x:A)} B(x) : \mathcal{U}_n$
- if, in addition, $a : A$ and $b : B(a)$, then $(a, b) : \sum_{(x:A)} B(x)$

If we have A and B as above, $C : (\sum_{(x:A)} B(x)) \to \mathcal{U}_m$, and

$$d : \prod_{(x:A)} \prod_{(y:B(x))} C((x, y))$$

we can introduce a defined constant

$$f : \prod_{(p:\sum_{(x:A)} B(x))} C(p)$$

with the defining equation

$$f((x, y)) :\equiv d(x, y).$$

Suspensions

It is a type ΣA defined by the following generators:

- a point $\mathsf{N} : \Sigma A$,
- a point $\mathsf{S} : \Sigma A$, and
- a function $\mathsf{merid} : A \to (\mathsf{N} =_{\Sigma A} \mathsf{S})$.

The recursion principle for ΣA says that given a type B together with

- points $n, s : B$ and
- a function $m : A \to (n = s)$,

we have a function $f : \Sigma A \to B$ such that $f(\mathsf{N}) \equiv n$ and $f(\mathsf{S}) \equiv s$, and for all $a : A$ we have $f(\mathsf{merid}(a)) = m(a)$.

Similarly, the induction principle says that given $P : \Sigma A \to \mathcal{U}$ together with

- a point $n : P(\mathsf{N})$,
- a point $s : P(\mathsf{S})$, and
- for each $a : A$, a path $m(a) : n =^{P}_{\mathsf{merid}(a)} s$,

there exists a function $f : \prod_{(x:\Sigma A)} P(x)$ such that $f(\mathsf{N}) \equiv n$ and $f(\mathsf{S}) \equiv s$ and for each $a : A$ we have $\mathsf{apd}_f(\mathsf{merid}(a)) = m(a)$.