En este capítulo, veremos algunas aplicaciones de HoTT en la topología algebraica. Analizaremos a mayor profundidad la estructura homotópica de los tipos, y veremos algunos resultados clásicos involucrando conceptos como el de contractibilidad, CW complejos, y el grupo fundamental.

0.1. Topología y tipos

Puesto que los tipos son tipos de homotopía de espacios topológicos, se puede formalizar varias propiedades clásicas que estos cumplen. Para comenzar, ahora podemos ver la demostración completa de la proposición mencionada previamente de que las familias de tipos son fibraciones.

Proposición 0.1.1. *Sea $P : B \to \mathcal{U}$ una familia de tipos sobre B, y $f, g : X \to B$ dos funciones tales que existe una homotopía h entre ellas. Si existe un levantamiento de f, $\tilde{f} : \prod_{(x:X)} P(f(x))$; entonces existe una homotopía $\tilde{h}$ entre $\tilde{f}$ y un levantamiento de g, tal que $\tilde{h}$ levanta a h,*

Demostración. Podemos definir una homotopía

$$\tilde{h} : \prod_{x:X} (f(x), \tilde{f}(x)) =_{\sum_{(x:X)} P(x)} \Big(g(x), \mathsf{tr}^P(h(x), \tilde{f}(x))\Big)$$

por $\tilde{h}(x) = \mathsf{pair}^=(h(x), \mathsf{refl})$. Para ver que esta levanta a H, necesitamos comprobar que

$$\prod_{x:X} \mathsf{pr}_1(\tilde{h}(x)) = h(x)$$

Sin embargo, por inducción, es fácil notar que $\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{pair}^=(p, q)) = p$, para todo p, q. Aplicando esto en la ecuación anterior, obtenemos el resultado deseado. □

Notamos que, con los conceptos ya introducidos, es fácil formalizara varios conceptos topológicos. Por ejemplo, podemos definir los conceptos de retracciones y secciones.

Definición 0.1.2. Una **retracción** es una función $r : A \to B$ tal que existe una función $s : B \to A$, su **sección**, y una homotopía $\epsilon : \prod_{(y:B)}(r(s(y)) = y)$. Cuando esto se de, decimos que B es un **retracto** de A.

Veamos cómo podemos hacer un análisis similar a otros conceptos.

0.2. n-tipos

En topología clásica, un espacio topológico es llamado contractible cuando la función identidad es homotópica a la imagen de un solo punto. Esto sugiere:

Definición 0.2.1. Un tipo A es **contractible** cuando posee un elemento $a : A$, llamado el **centro de contracción**, tal que $a = x$, para todo $x : A$. Es decir, definimos la propiedad de ser contractible como:

$$\mathsf{isContr}(A) :\equiv \sum_{(a:A)} \prod_{(x:A)} (a = x).$$

Ejemplo 0.2.2. El tipo $\mathbf{1}$ es contractible, pues para mostrar que $\prod_{(x:\mathbf{1})}(\star = x)$, podemos asumir que x es $\star$, y podemos tomar el camino constante.

Con este resultado, podemos demostrar que el tipo $\mathbf{1}$ es un objeto terminal.

Proposición 0.2.3. *Para todo tipo* C *y función* $f : \mathsf{C} \to \mathbf{1}$*, se tiene que* $f = !\mathbf{1}_{\mathsf{C}}$.

Demostración. Para todo $x : \mathsf{C}$, se tiene $f(x) = \star$, por el resultado previo. Por extensionalidad de funciones, entonces $f = !\mathbf{1}_{\mathsf{C}}$. □

Desde una perspectiva lógica, la contractibilidad indica que todos los elementos del tipo son iguales a un elemento en específico. Topológicamente, tenemos una función continua que lleva cualquier punto $x : A$ hacia a. Esto sugiere que el tipo A es equivalente al tipo $\mathbf{1}$.

Proposición 0.2.4. *Un tipo* A *es contractible sí y solo sí este es equivalente al tipo* $\mathbf{1}$.

Demostración. Sea $(c, p) : \mathsf{isContr}(A)$, podemos definir $f : A \to \mathbf{1}$ por $f(x) = \star$. En sentido contrario, tenemos $g : \mathbf{1} \to A$ dado por $g(y) = c$. Ahora, por un lado, realizando inducción en $\mathbf{1}$, vemos que para todo $y : \mathbf{1}$, $f(g(y)) = \star$. Por otro lado, para todo $x : A$, tenemos $g(f(x)) = c = x$, donde la segunda igualdad se da por $p(x)$. Juntando estos resultamos, obtenemos que $A \simeq \mathbf{1}$.

Para mostrar la recíproca, supongamos que $A \simeq \mathbf{1}$, y sean $f : A \to \mathbf{1}$ y $g : \mathbf{1} \to A$ las quasi-inversas. Por el ejemplo previo, tenemos que $\star = f(x)$. Así, se tiene:

$$g(\star) = g(f(x)) = x,$$

con lo que $g(\star)$ es el centro de contracción. □

Veamos que este, al igual que en MC, esta propiedad se preserva bajo varias operaciones, como productos y retracciones.

Proposición 0.2.5. *Sea* $B : A \to \mathcal{U}$ *una familia de tipos. Si* A *es contractible y para todo* $x : A$*,* $B(x)$ *es contractible, entonces* $\sum_{(x:A)} B(x)$ *también lo es.*

Demostración. Sea $a_0 : A$ el centro de contracción de A, y sea $b_0 : B(a_0)$ el centro de contración de $B(a_0)$. Mostraremos que (a_0, b_0) es un centro de contracción. Dado $(a, b) : \sum_{(x:A)} B(x)$, necesitamos mostrar camino $p : a_0 = a$ tal que $\mathsf{tr}^B(p, b_0) = b$. Por un lado, este p existe por contractibilidad de A. Por otro lado, tenemos:

$$\begin{aligned}
\mathsf{tr}^B(p, b_0) &= \mathsf{tr}^B(p, \mathsf{tr}^B(p^{-1}, b)) && \text{Por contractibilidad de } B(a_0) \\
&= \mathsf{tr}^P(p^{-1} \cdot p, b) && \text{Por el Lema ??} \\
&= \mathsf{tr}^P(\mathsf{refl}_a, b) \\
&\equiv b
\end{aligned}$$

□

Corolario 0.2.6. *La contractibilidad se preserva bajo productos.*

Demostración. Aplicación directa de la proposición anterior en el tipo $\lambda(x : A).\, B$. □

Proposición 0.2.7. *Sea B contractible y sea $r : B \to A$ una retracción. Entonces A es contractible.*

Demostración. Sea $b : B$ el centro de contracción de B, y sea $s : A \to B$ la sección correspondiente de r. Mostraremos que $r(b_0)$ es un centro de contracción de A, en efecto, tenemos:

$$\begin{aligned} r(b_0) &= r(s(a)) && \text{Por contractibilidad de } B \\ &= a && \text{Por ser } s \text{ la sección de } r \end{aligned}$$

□

Consideremos el tipo

$$\prod_{x,y:A} \mathsf{isContr}(x =_A y) \equiv \prod_{(x,y:A)} \sum_{(p:x=y)} \prod_{(q:x=y)} (p = q) \tag{1}$$

Este tipo indica que, para todo par de elementos de A, el tipo de caminos entre ellos es contractible. En particular, para todo $x, y : A$, tenemos un camino $p : x = y$ entre estos. Para ver que esta condición es también suficiente, necesitaremos un lema previo.

Lema 0.2.8. *Sea A un tipo, $a, x_1, y_1 : A$ y $p : x_1 = x_2, q : a = x_1$ caminos. Entonces*

$$\mathsf{tr}^{\lambda x \to a = x}(p, q) = q \cdot p$$

Demostración. Aplicando inducción a ambos caminos, ambos lados de la igualdad se reducen a refl_a. □

Nótese que este lema corresponde a la acción functorial (covariante) del functor $\mathrm{Hom}(a, -)$ mencionado en el Ejemplo **??**. Existe un resultado análogo para $\mathsf{tr}^{\lambda x \to x = a}$, correspondiente a $\mathrm{Hom}(-, a)$, el cual no necesitaremos.

Proposición 0.2.9. *Sea A un tipo. El tipo* (1) *está habitado sí y solo sí para todo $x, y : A$, tenemos que $x = y$.*

Demostración. Ya mostramos un lado de la implicación, veamos el caso de la recíproca. Por hipótesis, tenemos una función $f : \prod_{(x,y:A)} x = y$. Consideremos la función $g : \prod_{(z:A)} x = z$ definida por $g(z) = f(x, z)$.

Mostraremos que $(g(x))^{-1} \cdot g(y)$ es el centro de contracción. En efecto, dado $p : x = y$ tenemos:

$$\begin{aligned} (g(x))^{-1} \cdot g(y) &= (g(x))^{-1} \cdot \mathsf{tr}^{\lambda z \to x = z}(p, g(x)) && \text{Por el Lema ??} \\ &= (g(x))^{-1} \cdot (g(x) \cdot p) && \text{Por el Lema 0.2.8} \\ &= p \end{aligned}$$

□

Ya habíamos encontrado esta propiedad previamente, en nuestra discusión del axioma K en la sección **??**, pues este axioma indica justamente que todos los elementos del tipo $x =_A y$ son iguales entre sí. Desde una perspectiva lógica, las proposiciones están caracterizadas por la propiedad de tener un valor de verdad, ser verdaderas o falsas. No es relevante qué elemento en particular tenemos de una proposición, por lo que podemos considerar a todas como iguales. Esto sugiere que el tipo (1) previo corresponde a la propiedad de ser una proposición.

Definición 0.2.10. Decimos que un tipo P es una **proposición simple** si es que el siguiente tipo está habitado.

$$\mathsf{isProp}(P) :\equiv \prod_{x,y:A} (x = y)$$

Ejemplo 0.2.11. Todo tipo contractible es una proposición simple. En efecto, sea $(c, p) : \mathsf{isContr}(A)$, entonces para todo $x, y : A$ tenemos $x = c = y$. En particular, el tipo $\mathbf{1}$ es una proposición simple.

Ejemplo 0.2.12. El tipo $\mathbf{0}$ es una proposición simple. Si tenemos $x, y : \mathbf{0}$ tenemos inmediatamente una contradicción, por lo que podemos derivar que el resultado deseado.

Ejemplo 0.2.13. Para toda $f : A \to B$, $\mathsf{isequiv}(f)$ es una proposición simple; este es justo uno de los requisitos que habíamos pedido de una noción correcta de equivalencia homotópica.

Ejemplo 0.2.14. Para todo $m, n : \mathbb{N}$, el tipo $m = n$ es una proposición simple. Puesto que $(m = n) \simeq (\mathsf{code}(m, n))$, es suficiente demostrar que $\mathsf{code}(m, n)$ es una proposición simple. Procedemos por inducción en m y n. Cuando ambos son 0, entonces $\mathsf{code}(0, 0) \equiv \mathbf{1}$, y ya hemos visto que esta es una proposición simple. Cuando uno es un sucesor y el otro no, $\mathsf{code}(m, n) \equiv \mathbf{0}$, y sabemos que este tipo también es una proposición simple. Finalmente, cuando tenemos $\mathsf{succ}(m)$ y $\mathsf{succ}(n)$, entonces $\mathsf{code}(\mathsf{succ}(m), \mathsf{succ}(n)) \equiv \mathsf{code}(m, n)$, por lo que podemos aplicar la hipótesis de inducción.

Podemos volver a iterar, y preguntarnos qué propiedad satisfacen aquellos tipos cuyos caminos son todos proposiciones simples; veremos que esta es justamente la propiedad de ser un conjunto.

Definición 0.2.15. Un tipo A es un **conjunto** si el siguiente tipo está habitado

$$\mathsf{isSet}(A) :\equiv \prod_{(x,y:A)} \prod_{(p,q:x=y)} (p = q)$$

Topológicamente, esta propiedad nos indica que todo par de caminos con los mismos extremos son homotópicos entre sí. Es decir, tienen el mismo tipo de homotopía que un espacio con la topología discreta. Así, lo único relevante en un conjunto son sus elementos diferentes, lo que captura la noción usual de MC.

Ejemplo 0.2.16. El tipo $\mathbf{0}$ es un conjunto, pues de $x, y : \mathbf{0}$ se deriva un absurdo.

Ejemplo 0.2.17. El tipo $\mathbb{N}$ es un conjunto, pues por el Ejemplo 0.2.14, para todo $m, n : \mathbb{N}$, $m = n$ es una proposición simple.

Nótese que con este tipo, podemos hacer definiciones como el tipo de todos los conjuntos:

$$\mathsf{Set}_{\mathcal{U}} :\equiv \sum_{A:\mathcal{U}} \mathsf{isSet}(A)$$

Es claro que podemos siguer iterando este tipo de preguntas al infinito. Podemos indigar respecto a todas estas propiedades, definiendo la siguiente familia de tipos.

Definición 0.2.18. Definimos ser un n-tipo, a través de recursión en los naturales de la siguiente forma

$$\begin{aligned}\mathsf{is}\text{-}(n-2)\text{-}\mathsf{type} &: \mathcal{U} \to \mathcal{U} \\ \mathsf{is}\text{-}(0-2)\text{-}\mathsf{type}(A) &:\equiv \mathsf{isContr}(A) \\ \mathsf{is}\text{-}(\mathsf{succ}(n)-2)\text{-}\mathsf{type}(A) &:\equiv \prod_{x,y:A} \mathsf{is}\text{-}(n-2)\text{-}\mathsf{type}(A)\end{aligned}$$

Así, los (-2)-tipos son los tipos contractibles, los (-1)-tipos las proposiciones simples, los 0-tipos los conjuntos, etc. La razón de empezar desde el -2 es porque ahora tenemos la siguiente descripción más sugerente: para $n : \mathbb{N}$, el tipo A es un n-tipo si solo tiene m-caminos no-triviales para $m \leq n$. Esta es una noción comúnmente estudiada en topología algebraica, llamada ahí el n-tipo de homotopía.

Dada esta nueva interpretación, nuestra intuición geométrica sugiere una serie de resultados. Mostraremos aquí algunos.

Proposición 0.2.19. *La jerarquía de n-tipos es cumulativa; es decir, si A es un n-tipo, también es un $\mathsf{succ}(n)$-tipo.*

Demostración. Procedemos por inducción en n. Para el caso base, sea A un tipo contractible, necesitamos mostrar que los caminos en A son contractibles. Sea $(c, p) : \mathsf{isContr}(A)$, y sean x, y en A. Primero, tenemos que $p(x)^{-1} \cdot (p(y)) : x = y$. Queremos mostrar que este es el centro de contracción. Dado otro camino $q : x = y$, por inducción, es suficiente mostrar que $p(x) \cdot (p(x))^{-1} = \mathsf{refl}_x$, lo que ya hemos visto previamente.

Para el caso general, si A es un $\mathsf{succ}(n)$-tipo, tenemos que $x = y$ es un n-tipo por definición, por lo que por la hipótesis de inducción concluimos que $x = y$ también es un $\mathsf{succ}(n)$-tipo □

Proposición 0.2.20. *Los n-tipos se preservan bajo retracciones. Es decir, para todo $n \geq -2$, si $r : B \to A$ es una retracción y B es un n-tipo, entonces A también es un n-tipo.*

Demostración. Realizaremos la prueba por inducción en n. El caso base, lo muestra la Proposición 0.2.7. Entonces, supongamos que B es un $\mathsf{succ}(n)$-tipo, y sean $s : A \to B$ la sección de r, y $\epsilon : r \circ s \sim \mathsf{id}$ la homotopía correspondiente.

Dados $a_1, a_2 : A$ arbitrarios, necesitamos mostrar que $a_1 = a_2$ es un n-tipo, pero por la hipótesis de inducción, es suficiente mostrar que tipo es un retracto de $s(a_1) = s(a_2)$. Realizaremos esto a continuación.

Para la sección, tomamos

$$\mathsf{ap}_s : (a_1 = a_2) \to (s(a_1) = s(a_2)),$$

mientras que la retracción $t : (s(a_1) = s(a_2)) \to (a_1 = a_2)$ la definimos por

$$t(q) :\equiv \epsilon_{a_1}^{-1} \cdot r(q) \cdot \epsilon_{a_2}.$$

Para concluir la prueba, necesitamos $t \circ \mathsf{ap}_{\sim}\mathsf{id}$; es decir, que para todo $p : a_1 = a_2$ se tenga que

$$\epsilon_{a_1}^{-1} \cdot r(s(p)) \cdot \epsilon_{a_2} = p,$$

pero esto es una consecuencia del Lema **??**. □

Como una consecuencia de esta proposición, podemos obtener el siguiente resultado sin usar univalencia.

Corolario 0.2.21. *Los n-tipos se preservan bajo equivalencias.*

Demostración. Una la existencia de una equivalencia $A \simeq B$ implica que A es una retracción de B, por lo que si B es un n-tipo, entonces A también lo es. □

Finalmente, presentamos la generalización de la Proposición 0.2.5.

Proposición 0.2.22. *Los n-tipos se preservan bajo sumas dependientes. Es decir, sea para todo $n \geq -2$, si A es un n-tipo y la familia de tipos $B : A \to \mathcal{U}$ satisface que $B(a)$ es un n-tipo para todo $a : A$, entonces $\sum_{(x:A)} B(x)$ es un n-tipo.*

Demostración. Realizaremos la prueba por inducción en n. El caso base, lo muestra la Proposición 0.2.5. Entonces, supongamos que A es un $\mathsf{succ}(n)$-tipo y $B(a)$ es un $\mathsf{succ}(n)$-tipo para todo $a : A$. Dados $(a_1, b_1), (a_2, b_2) : \sum_{(x:A)} B(x)$ arbitrarios, necesitamos mostrar que $(a_1, b_1) = (a_2, b_2)$ es un n-tipo. Pero por la caracterización de caminos dependientes, tenemos:

$$((a_1, b_1) = (a_2, b_2)) \simeq \sum_{p:a_1=a_2} (\mathsf{tr}^B(p, b_1) = b_2)$$

Pero el tipo de la derecha es un n-tipo por la hipótesis de inducción, por lo que, por el Corolario 0.2.21, el tipo de la izquierda también lo es. □

0.3. Tipos Inductivos Superiores

Como ya mencionamos en la Sección **??**, los tipos que hemos introducidos hasta ahora, pueden ser considerados como aquellos libremente generados por sus *constructores*, aquellas constantes y funciones que nos permiten formar elementos del tipo. Así, por ejemplo, los naturales están libremente generados por la existencia de un elemento $0 : \mathbb{N}$ y la función $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$.

Podemos expandir este procedimiento para que los constructores incluyan no solo elementos del tipo, sino también caminos entre los elementos. Aquellos tipos que incluyen constructores de caminos son llamados **Tipos Inductivos Superiores**, o **HITs**, por su nombre en inglés, *Higher Inductive Types.* Veamos uno de estos.

Definición 0.3.1. El **intervalo**, denotado I, es el tipo generado por

- un punto $0_I : I$,
- un punto $1_I : I$, y
- un camino $\mathsf{seg} : 0_I = 1_I$.

Este tipo representa un camino abstracto; es decir, el tipo de homotopía de un intervalo cerrado en los reales. Nótese que la definición logra esto sin hacer ninguna mención a toda la maquinaria usual de MC (distancia, bolas, topología, clases de equivalencia, etc).

El principio de recursión de los tipos indica que estos pueden ser mapeados hacia tipos que comparten su estructura interna. En efecto, existe un algoritmo que permite deducir el principio de recursión (y el de inducción), dado los constructores de un tipo. No abordaremos este tema, y daremos estos principios para los tipos que usaremos.

En el caso del tipo del intervalo, el principio de recursión dice que dado un tipo B junto con

- un punto $b_0 : B$,
- un punto $b_1 : B$, y
- un camino $s : b_0 = b_1$,

existe una única función $f : I \to B$ tal que $f(0_I) \equiv b_0, f(1_I) \equiv b_1$, y $f(\mathsf{seg}) = s$. Nótece que las primeras dos igualdades son por definición, mientras que la tercera de estas es una igualdad proposicional.

El principio de inducción de un tipo, como ya hemos mencionado, es el principio de recursión generalizado para familias dependientes. Entonces, para el caso dice que dado una familia de tipos $P : I \to \mathcal{U}$, junto con

- un punto $b_0 : P(0_I)$,
- un punto $b_1 : P(1_I)$, y
- un camino levantado $s : P(b_0) =^P_{\mathsf{seg}} P(b_1)$,

existe una única función $f : \prod_{(x:I)} P(x)$ tal que $f(0_I) \equiv b_0, f(1_I) \equiv b_1$, y $\mathsf{apd}_f(\mathsf{seg}) = s$. Con este principio, podemos demostrar que, como se esperaría, este tipo es contractible.

Proposición 0.3.2. *El tipo I es contractible.*

Demostración. Mostraremos que 0_I es un centro de contracción. Así, necesitamos una función $\prod_{(x:I)}(0_I = x)$. Por el principio de inducción, podemos definir f por:

$$\begin{aligned} f(0_I) &:\equiv \mathsf{refl}_{0_I} : 0_I = 0_I, \\ f(1_I) &:\equiv \mathsf{seg} \;\; : 0_I = 1_I. \end{aligned}$$

junto con el hecho de que existe un camino $p : \mathsf{refl}_{0_I} =^{\lambda x.\, 0_I = x}_{\mathsf{seg}} \mathsf{seg}$. Pero por el Lema 0.2.8, esto es equivalente a un camino $\mathsf{refl}_{0_I} \cdot \mathsf{seg} = \mathsf{seg}$, el cual tenemos por el Lema **??**. □

Por este resultado, junto con la Proposición 0.2.4, el tipo I es equivalente al tipo $\mathbf{1}$. A pesar de esto, este tipo igual es interesante, pues nos permite realizar otros tipos de argumentos.

Teorema 0.3.3. *Sean $f, g : A \to B$ funciones tales que para todo $x : A$ se tiene que $f(x) = g(x)$. Entonces, $f = g$.*

Demostración. Por recursión, podemos definir una función $\alpha : A \to (I \to B)$ por

$$\begin{aligned} \alpha(x)(0_I) &:\equiv f(x) : B, \\ \alpha(x)(1_I) &:\equiv g(x) : B. \end{aligned}$$

notando que por suposición, tenemos un camino $p : f(x) = g(x)$. Ahora, podemos invertir el orden de las variables, generando una función

$$\beta :\equiv \lambda(x : I).\, \lambda(y : A).\, \alpha(y)(x) : I \to (A \to B).$$

Finalmente, vemos que $\beta(seg) : \beta(0_I) = \beta(1_I)$, pero $\beta(0_I) \equiv f$ y $\beta(1_I) \equiv g$. □

Otro HIT, quizás más interesante, es el círculo, el cual estudiaremos a más detalle en la próxima sección.

Definición 0.3.4. El **círculo**, denotado $\mathbb{S}^1$, es el tipo generado por

- un punto $\mathsf{base} : \mathbb{S}^1$, y
- un camino $\mathsf{loop} : \mathsf{base} = \mathsf{base}$.

Su principio de recursión indica que dados

- un punto $b : B$, y
- un camino $\ell : b = b$,

existe una única función $f : \mathbb{S}^1 \to B$ tal que $f(\mathsf{base}) \equiv b$ y $f(\mathsf{loop}) = \ell$. El principio de inducción dice que dada una familia de tipos $P : \mathbb{S}^1 \to \mathcal{U}$, junto con

- un punto $b : B(\mathsf{base})$, y
- un camino $\ell : b =^B_{\mathsf{loop}} b$,

existe una única función $f : \prod_{(x:\mathbb{S}^1)} B(x)$ tal que $f(\mathsf{base}) \equiv b$ y $\mathsf{apd}_f(\mathsf{loop}) = \ell$.

Similarmente, usando esta misma técnica, podemos definir la esfera.

Definición 0.3.5. La **esfera** $\mathbb{S}^2$ es el tipo generado por

- un punto $\mathsf{base} : \mathbb{S}^2$, y
- un 2-camino $\mathsf{surf} : \mathsf{refl}_{\mathsf{base}} = \mathsf{refl}_{\mathsf{base}}$,

Este tipo de definiciones es análogo a la construcción de algunos espacios como CW complejos. Vamos un último ejemplo.

Definición 0.3.6. Podemos definir al **toro** T^2 como el tipo generado por

- un punto $b : T^2$,
- un camino $p : b = b$,
- otro camino $q : b = b$, y
- un 2-camino camino $t : p \cdot q = q \cdot p$,

Esta definición corresponde al toro como un cuadrado con los lados opuestos identificados. Nótese la mayor elegancia y simplicidad de esta construcción, que evita apelar a nociones técnicas como lo es la topología cociente.

Los HITs no solo sirven para crear espacios topológicos (hasta homotopía), sino también permite la implementación de algunas operaciones sobre tipos, veamos un ejemplo.

Definición 0.3.7. Dado un tipo A, podemos definir su **0-truncación** $\|A\|_0$ como el tipo generado por

- una función $|-|_0 : A \to \|A\|_0$, y
- para todo par de puntos $x, y : \|A\|_0$ y par de caminos $p, q : x = y$, una igualdad $p = q$.

Su principio de recursión indica que dados

- una función $g : A \to B$, y

- para todo par de puntos $x, y : B$ y par de caminos $p, q : x = y$, una igualdad $p = q$,

existe una única función $f : \|A\|_0 \to B$ tal que $f(|x|_0) \equiv g(x)$, para todo $x : A$, y ap_f lleva el camino de $p = q$ en $\|A\|_0$ en el camino especificado $f(p) = f(q)$ en B.

Este tipo efectivamente trivializa los 1-caminos, pues todos se vuelven homotópicos entre sí. Por otro lado, esto es lo único que hace, como uno esperaría, deja a los conjuntos intactos.

Proposición 0.3.8. *Sea A un conjunto, entonces $\|A\|_0 = A$.*

Demostración. Por univalencia, es suficiente demostrar que $\|A\|_0 \simeq A$. Por un lado, tenemos una función $f : \|A\|_0 \to A$ definida por recursión, con $f(|x|_0) \equiv x$. Por otro lado, tomamos $|-|_0 : A \to \|A\|_0$ como la quasi-inversa.

Por definición de f, vemos que $f \circ |-|_0 \sim \mathsf{id}$. Para ver que $|-|_0 \circ f \sim \mathsf{id}$, por unicidad de la función proveniente de recursión, solo necesitamos comprobar que

$$(|-|_0 \circ f \circ |-|_0)(x) = |x|_0,$$

pero nuevamente esto se da por definición de f. □

Existen n-truncaciones para $n \geq -1$, las cuales "truncan" A hacia un n-tipo, pero no abordaremos estas construcciones.

Finalmente, los HITs permiten la definición de tipos de una manera más concisa. Por ejemplo, el tipo de los enteros $\mathbb{Z}$ generalmente se define como un cociente de $\mathbb{N} \times \mathbb{N}$; sin embargo, la siguiente descripción [1] es más útil (y más elegante).

Definición 0.3.9. El tipo de los **enteros** $\mathbb{Z}$ es el tipo generado por

- un elemento $0_{\mathbb{Z}} : \mathbb{Z}$
- una equivalencia $\mathsf{succ}_{\mathbb{Z}} : \mathbb{Z} \simeq \mathbb{Z}$,

junto con el hecho de que $\mathbb{Z}$ es un conjunto.

Aquí, la equivalencia $\mathsf{succ}_{\mathbb{Z}}$ corresponde al hecho de que la función sucesor posee una quasi-inversa, la función predecesor.

El principio de recursión de $\mathbb{Z}$ indica que dados

- un punto $b : B$, y
- una equivalencia $s : B \simeq B$,

existe una única función $\mathbb{Z} \to B$ tal que $f(0_{\mathbb{Z}}) \equiv b$ y para todo $x : \mathbb{Z}$, $f(\mathsf{succ}_{\mathbb{Z}}(x)) = s(f(x))$.

El principio de inducción, lo omitimos, pues no lo necesitaremos.

0.4. El grupo fundamental del círculo

El grupo fundamental $\pi_1(X)$ permite generar un objeto algebraico, a partir de las operaciones de concatenación de los 1-caminos en X. Sin embargo, esto se puede generalizar para n-caminos, para $n \geq 1$.

Definición 0.4.1. El espacio de n-caminos cerrados en un tipo A con base $a : A$, es el tipo

$$\Omega^0(A, a) :\equiv (A, a)$$
$$\Omega^{\mathsf{succ}(n)}(A, a) :\equiv \Omega^n\left((a =_A a), \mathsf{refl}_a\right)$$

Definición 0.4.2. Sea $n \geq 1$ y sea A un tipo con $a : A$, definimos el **n-ésimo grupo de homotopía** de A centrado en a como el tipo

$$\pi_n(A, a) :\equiv \|\mathsf{pr}_1\left(\Omega^n(A, a)\right)\|_0$$

El objetivo de esta sección es demostrar que $\pi_1(\mathbb{S}^1, \mathsf{base}) = \mathbb{Z}$. De hecho, demostraremos algo aún más fuerte, que $(\mathsf{base} = \mathsf{base}) \simeq \mathbb{Z}$. Comencemos con el regreso de las quasi-inversas.

Lema 0.4.3. *Existe una función* $\hat{\mathsf{loop}} : \mathbb{Z} \to (\mathsf{base} = \mathsf{base})$.

Demostración. Por el principio de recursión de los enteros, necesitamos un elemento de $\mathsf{base} = \mathsf{base}$, el cual escogemos que sea $\mathsf{refl}_{\mathsf{base}}$, junto con una equivalencia $s : (\mathsf{base} = \mathsf{base}) \simeq (\mathsf{base} = \mathsf{base})$. Definiremos esta equivalencia como la asociada al siguiente par de quasi-inversas, $f(p) :\equiv p \cdot \mathsf{loop}$ y $g(p) :\equiv p \cdot \mathsf{loop}^{-1}$.

$$f(g(p)) \equiv p \cdot \mathsf{loop}^{-1} \cdot \mathsf{loop} = p$$
$$g(f(p)) \equiv p \cdot \mathsf{loop} \cdot \mathsf{loop}^{-1} = p$$

□

Esta función lleva al $0_{\mathbb{Z}}$ al camino constante, lleva a los enteros positivos n a n concatenaciones de loop, y enteros negativos $-n$ a N concatenaciones de loop^{-1}, de igual manera que en la demostración usual de MC.

Para generar una función $(\mathsf{base} = \mathsf{base}) \to \mathbb{Z}$ utilizaremos el método encode-decode, caracterizando el tipo de caminos $(\mathsf{base} = x)$, para $x : \mathbb{S}^1$.

Definición 0.4.4. Definimos la función $\mathsf{Cover} : \mathbb{S}^1 \to \mathcal{U}$ por recursión en el círculo:

$$\mathsf{Cover}(\mathsf{base}) :\equiv \mathbb{Z}$$
$$\mathsf{Cover}(\mathsf{loop}) := \mathsf{ua}(\mathsf{succ}_{\mathbb{Z}})$$

Así, Cover representa el espacio de cubrimiento usual del círculo. Podemos realizar cálculos con este tipo, por ejemplo, que transportando un natural a lo largo de loop es lo mismo que sumarle uno.

Lema 0.4.5. *Para todo $x : \mathbb{Z}$, se tiene que*

$$\mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}, x) = \mathsf{succ}_{\mathbb{Z}}(x)$$

Demostración. Tenemos que

$$\begin{aligned}
\mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}, x) &\equiv \mathsf{tr}^{\mathsf{id}\circ\mathsf{Cover}}(\mathsf{loop}, x) && \text{(Def. de id)}\\
&= \mathsf{tr}^{\mathsf{id}}(\mathsf{Cover}(\mathsf{loop}), x) && \text{(Lema ??)}\\
&= \mathsf{tr}^{\mathsf{id}}(\mathsf{ua}(\mathsf{succ}_{\mathbb{Z}}), x) && \text{(Def. de Cover)}\\
&\equiv \mathsf{pr}_1(\mathsf{idtoeqv}(\mathsf{ua}(\mathsf{succ}_{\mathbb{Z}}))) && \text{(Def. de idtoeqv)}\\
&= \mathsf{succ}_{\mathbb{Z}} && \text{(Def. de ua)}
\end{aligned}$$

□

Procederemos a demostrar ahora la equivalencia $(\mathsf{base} = x) \simeq \mathsf{Cover}(x)$, para todo $x : \mathbb{S}^1$. Tenemos que generar las dos quasi-inversas y mostrar que sus composiciones son homotópicas a la identidad.

Lema 0.4.6. *Tenemos una función*

$$\mathsf{encode} : \prod_{x:\mathbb{S}^1} (\mathsf{base} = x) \to \mathsf{Cover}(x)$$

Demostración. Definimos encode por $\mathsf{encode}(x, p) :\equiv \mathsf{tr}^{\mathsf{Cover}}(p, 0_{\mathbb{Z}})$. □

Intuitivamente, encode toma un camino $\mathsf{base} = x$ y lo lleva al tipo asociado transportando $0_{\mathbb{Z}}$.

Lema 0.4.7. *Tenemos una función*

$$\mathsf{decode} : \prod_{x:\mathbb{S}^1} \mathsf{Cover}(x) \to (\mathsf{base} = x)$$

Demostración. Definimos decode usando el principio de inducción sobre la familia de tipos

$$C :\equiv \lambda(x : \mathbb{S}^1).\ (\mathsf{Cover}(x) \to (\mathsf{base} = x))$$

Para esto, primero necesitamos un elemento de $C(\mathsf{base})$, tomaremos a $\mathsf{loop}\hat{}$ como este. Para completar la inducción necesitamos un camino $\mathsf{loop}\hat{} =^{C}_{\mathsf{loop}} \mathsf{loop}\hat{}$. Pero observamos que

$$\begin{aligned}
&\mathsf{tr}^{(\lambda x.\,(\mathsf{Cover}(x) \to (\mathsf{base}=x)))}(\mathsf{loop}, \mathsf{loop}\hat{}) \\
&\quad = \mathsf{tr}^{\mathsf{base}=-}(\mathsf{loop}, -) \circ \mathsf{loop}\hat{} \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}^{-1}, -) && \text{(Lema ??)} \\
&\quad = (- \cdot \mathsf{loop}) \circ \mathsf{loop}\hat{} \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}^{-1}, -) && \text{(Lema 0.2.8)} \\
&\quad = \mathsf{loop}\hat{} \circ \mathsf{succ}_{\mathbb{Z}} \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}^{-1}, -) && \text{(Def. de } \mathsf{loop}\hat{}) \\
&\quad = \mathsf{loop}\hat{} \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}, -) \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}^{-1}, -) && \text{(Lema 0.4.5)} \\
&\quad = \mathsf{loop}\hat{} \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}^{-1} \cdot \mathsf{loop}, -) && \text{(Lema ??)} \\
&\quad = \mathsf{loop}\hat{} \circ \mathsf{tr}^{\mathsf{Cover}}(\mathsf{refl}_{\mathsf{base}}, -) && \text{(Lema ??)} \\
&\quad = \mathsf{loop}\hat{} && \text{(Def. de } \mathsf{tr})
\end{aligned}$$

□

Intuitivamente, decode toma un elemento de $\mathsf{Cover}(x)$ y lo lleva al tipo asociado inducido por $\mathsf{loop}\hat{}$.

Ahora, la primera de las dos homotopía requeridas es inmediata.

Lema 0.4.8. *Para todo $x : \mathbb{S}^1$ y $p : \mathsf{base} = x$, tenemos que*

$$\mathsf{decode}(x, \mathsf{encode}(x, p)) = p$$

Demostración. Por inducción, podemos asumir que p es $\mathsf{refl}_{\mathsf{base}}$, pero entonces:

$$\begin{aligned}
\mathsf{decode}(\mathsf{base}, \mathsf{encode}(\mathsf{base}, \mathsf{refl}_{\mathsf{base}})) &\equiv \mathsf{decode}(\mathsf{base}, \mathsf{tr}^{\mathsf{Cover}}(\mathsf{refl}_{\mathsf{base}}, 0_{\mathbb{Z}})) \\
&\equiv \mathsf{decode}(\mathsf{base}, 0_{\mathbb{Z}}) \\
&\equiv \mathsf{loop}\hat{}(0_{\mathbb{Z}}) \\
&\equiv \mathsf{refl}_{\mathsf{base}}
\end{aligned}$$

□

La otra homotopía es más complicada, y necesitaremos uno lemas previos.

Lema 0.4.9. *Sea $f : \mathbb{Z} \to \mathbb{Z}$ una función tal que $f(0_\mathbb{Z}) = 0_\mathbb{Z}$ y tal que*

$$f \circ \mathsf{succ}_\mathbb{Z} \sim \mathsf{succ}_\mathbb{Z} \circ f,$$

entonces $f \sim \mathsf{id}$.

Demostración. Nótese que la función identidad cumple que

$$\mathsf{id} \circ \mathsf{succ}_\mathbb{Z} \sim \mathsf{succ}_\mathbb{Z} \circ \mathsf{id},$$

por lo que, por la unicidad de la función preveniente de recursión, $f \sim \mathsf{id}$. □

Este resultado indica que la identidad es la única función que conmuta con la función sucesor.

Lema 0.4.10. *Para todo $x : \mathbb{S}^1$, $p : x = \mathsf{base}$ y $y : \mathsf{Cover}(x)$ se tiene que*

$$\mathsf{tr}^{\mathsf{Cover}}(p \cdot \mathsf{loop}, y) = \mathsf{succ}_\mathbb{Z}(\mathsf{tr}^{\mathsf{Cover}}(p, y))$$

Demostración. Por inducción, podemos asumir que p es $\mathsf{refl}_{\mathsf{base}}$. Entonces, se tiene que

$$\mathsf{tr}^{\mathsf{Cover}}(\mathsf{refl}_{\mathsf{base}} \cdot \mathsf{loop}, y) = \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}, y) = \mathsf{succ}_\mathbb{Z}(y).$$

□

Este resultado indica cuando transportamos Cover a lo largo de un camino $p \cdot \mathsf{loop}$, esto es lo mismo que haber transportado a lo largo de p, y luego sumarle 1.

Lema 0.4.11. *Para todo $x : \mathbb{Z}$, se tiene que*

$$\mathsf{encode}(\mathsf{base}, \mathsf{loop}\hat{}(x)) = x$$

Demostración. Por el Lema 0.4.9, es suficiente mostrar que la función

$$h = \lambda(x : \mathbb{Z}).\, \mathsf{encode}(\mathsf{base}, \mathsf{loop}\hat{}(x))$$

satisface que evaluada en $0_\mathbb{Z}$ es $0_\mathbb{Z}$, y que conmuta con $\mathsf{succ}_\mathbb{Z}$. Por un lado, vemos que

$$h(0_\mathbb{Z}) \equiv \mathsf{encode}(\mathsf{base}, \mathsf{refl}_{\mathsf{base}}) \equiv 0_\mathbb{Z}$$

Por otro lado, tenemos que

$$\begin{aligned}
h(\mathsf{succ}_\mathbb{Z}(x)) &\equiv \mathsf{encode}(\mathsf{base}, \mathsf{loop}\hat{}(\mathsf{succ}_\mathbb{Z}(x))) && \text{(Def. } h\text{)} \\
&\equiv \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}\hat{}(\mathsf{succ}_\mathbb{Z}(x)), 0_\mathbb{Z}) && \text{(Def. } \mathsf{encode}\text{)} \\
&= \mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}\hat{}(x) \cdot \mathsf{loop}, 0_\mathbb{Z}) && \text{(Def. } \mathsf{loop}\hat{}\text{)} \\
&= \mathsf{succ}_\mathbb{Z}(\mathsf{tr}^{\mathsf{Cover}}(\mathsf{loop}\hat{}(x), 0_\mathbb{Z})) && \text{(Lema 0.4.10)} \\
&\equiv \mathsf{succ}_\mathbb{Z}(\mathsf{encode}(\mathsf{base}, \mathsf{loop}\hat{}(x))) && \text{(Def. } \mathsf{encode}\text{)} \\
&\equiv \mathsf{succ}_\mathbb{Z}(h(x)) && \text{(Def. } h\text{)}
\end{aligned}$$

□

Este resultado indica que transportar un el entero asociado a $x : \mathbb{Z}$ concatenaciones del camino loop, es x mismo.

Con todos estos lemas, procedemos a mostrar la última homotopía.

Lema 0.4.12. *Para todo* $x : \mathbb{S}^1$ *y* $p : \mathsf{Cover}(x)$*, se tiene que*

$$\mathsf{encode}(x, \mathsf{decode}(x, p)) = p$$

Demostración. Sea

$$C = \lambda(x : \mathbb{S}^1). \prod_{p : \mathsf{Cover} x} \mathsf{encode}(x, \mathsf{decode}(x, p)) = p.$$

El lema es equivalente a mostrar una función $\lambda(x : \mathbb{S}^1). C(x)$. Probaremos la existencia de esta función usando el principio de inducción del círculo. Primero debemos dar un camino

$$q : \mathsf{encode}(\mathsf{base}, \mathsf{loop}\hat{}(x)) = x,$$

el cual tenemos por el lema previo. Ahora, necesitamos un camino $q =^{C}_{\mathsf{loop}} q$. Pero por el Lema **??**, es suficiente que para todo camino $\beta : a_1 =^{\mathsf{Cover}}_{\mathsf{loop}} a_2$ se tenga que

$$q(a_1) =^{\lambda((x,y) : \sum_{(x:\mathbb{S}^1)} \mathsf{Cover}(x)).\, \mathsf{encode}(x, \mathsf{decode}(x,y)) = y}_{\mathsf{pair}^{=}(\mathsf{loop}, \alpha)} q(a_2)$$

Pero puesto que $\mathbb{Z}$ es un conjunto, ambos caminos son iguales inmediatamente. □

Con todos estos resultados, obtenemos una caracterización del tipo de caminos $\mathsf{base} = x$.

Proposición 0.4.13. *Para todo* $x : \mathbb{S}^1$*, tenemos que* $\mathsf{base} = x \simeq \mathsf{Cover}(x)$*.*

Aplicando esto en $x :\equiv \mathsf{base}$, por univalencia obtenemos el siguiente corolario.

Corolario 0.4.14. *El tipo* $\mathsf{base} = \mathsf{base}$ *es igual a el tipo de los enteros* $\mathbb{Z}$*.*

Finalmente, podemos calcular el grupo fundamental del círculo.

Corolario 0.4.15. *El grupo fundamental del círculo es* $\mathbb{Z}$*.*

Demostración. Tenemos que $\|\mathbb{Z}\|_0 = \mathbb{Z}$ por la proposición 0.3.8. Entonces, como $\mathsf{base} = \mathsf{base}$ es $\mathbb{Z}$, obtenemos $\pi_1(\mathbb{S}^1) = \mathbb{Z}$. □