Symmetric Monoidal Smash Products in HoTT

Axel Ljungström[1]

Stockholm University, Department of Mathematics
axel.ljungstrom@math.su.se

1 Introduction

In his 2016 proof of $\pi_4(S^3) \cong \mathbb{Z}/2\mathbb{Z}$ in Homotopy Type Theory (HoTT), Brunerie [Bru16] crucially uses—but never proves in detail—that the smash product is symmetric monoidal. Due to the vast amount of path algebra involved when reasoning about smash products, this has since remained open. While it turns out that smash products are not needed for Brunerie's proof [LM23; BLM22], the problem is still interesting in its own right. Several attempts have been made at salvaging the situation. Floris Van Doorn [Doo18] came very close by considering an argument using closed monoidal categories but left a gap where the path algebra got too technical. To be more precise, van Dorn never verified that the equivalence

$$(A \wedge B \to_\star C) \simeq (A \to_\star (B \to_\star C))$$

is a pointed natural equivalence. Another line of attack by Cavallo and Harper [CH20; Cav21] is the addition of parametricity to the type theory which leads to a rather ingenious proof of the theorem. This, of course, happens at the cost of making the type theory more complicated. Yet another solution was studied by Brunerie [Bru18] who used Agda meta-programming to generate the relevant proofs. Possible philosophical objections to such a solution aside, Brunerie's generated proof of the pentagon identity failed to type-check due to its eating up too much memory.

In this paper, we provide another solution. We introduce a heuristic for reasoning about functions defined over iterated smash products which vastly reduces the complexity of identity proofs. We use this to give a complete proof of the fact that the smash product is symmetric monoidal. While all key results have been formalised in Cubical Agda, we present them here in Book HoTT.

2 Background

Let us briefly introduce the key concepts of this paper: symmetric monoidal (wild) categories and smash products. We assume familiarity with HoTT and refer to the HoTT Book [UF13] for the basic constructions and definitions used here.

2.1 Symmetric monoidal wild categories

To make the statements in this paper somewhat more compact, we will use wild categories. These are defined to be just like categories but without any h-level assumptions [CK17].

Definition 1 (Wild categories)**.** *A wild category is a category with a* ***type*** *of objects and* ***types*** *of morphisms.*

The difference between a wild category and a category is that in the latter we ask for *sets* of objects and morphisms. While the definition of wild categories, in general, is much less well-behaved, it is general enough to capture what we will need in this paper. Here, the wild category of interest is that of pointed types.

Proposition 1. *Let* $\mathsf{Type}_\star$ *denote the universe of pointed types (at some universe level). This universe forms a wild category with* $\mathsf{Type}_\star[A, B] := (A \to_\star B)$*, i.e. with pointed functions as morphisms.*

The main goal of this paper is to show that $\mathsf{Type}_\star$ is not only a wild category but a *symmetric monoidal* wild category, so let us define this.

Definition 2 (Monoidal wild categories)**.** *A monoidal wild category is a wild category* M *equipped with*

- *a functor* $\otimes : M \times M \to M$
- *a unit, i.e. an element* $I : M$ *equipped with natural isomorphisms* $\lambda_A : I \otimes A \cong A$ *and* $\rho_A : A \otimes I \cong A$
- *a family of isomorphisms* $\alpha_{A,B,C} : ((A \otimes B) \otimes C) \cong (A \otimes (B \otimes C))$ *natural in all arguments such that*
  - *the* triangle indentity *holds, i.e. the following diagram commutes*

$$\begin{tikzcd}
(A \otimes I) \otimes B \arrow[rr, "\alpha_{A,I,B}"] \arrow[dr, "\rho_A \otimes 1_B"'] & & A \otimes (I \otimes B) \arrow[dl, "1_A \otimes \lambda_B"] \\
& A \otimes B &
\end{tikzcd}$$

  - *the* pentagon indentity *holds, i.e. the following diagram commutes*

$$\begin{tikzcd}
& ((A \otimes B) \otimes C) \otimes D \arrow[dl, "\alpha_{A,B,C} \otimes 1_D"'] \arrow[dr, "\alpha_{A \otimes B,C,D}"] & \\
(A \otimes (B \otimes C)) \otimes D \arrow[d, "\alpha_{A,B \otimes C,D}"'] & & (A \otimes B) \otimes (C \otimes D) \arrow[d, "\alpha_{A,B,C \otimes D}"] \\
A \otimes ((B \otimes C) \otimes D) \arrow[rr, "1_A \otimes \alpha_{B,C,D}"] & & A \otimes (B \otimes (C \otimes D))
\end{tikzcd}$$

Definition 3 (Symmetric monoidal wild categories)**.** *A symmetric monoidal wild category is a monoidal wild category equipped with a family of isomorphisms* $\beta_{A,B} : A \otimes B \cong B \otimes A$*, natural in both arguments, such that*

- $\beta_{B,A} \circ \beta_{A,B} = 1_{A \otimes B}$
- *The* hexagon identity *holds, i.e. the following diagram commutes*

$$\begin{tikzcd}
(A \otimes B) \otimes C \arrow[r, "\beta_{A,B} \otimes 1_C"] \arrow[d, "\alpha_{A,B,C}"'] & (B \otimes A) \otimes C \arrow[d, "\alpha_{B,A,C}"] \\
A \otimes (B \otimes C) \arrow[d, "\beta_{A,B \otimes C}"'] & B \otimes (A \otimes C) \arrow[d, "1_B \otimes \beta_{A,C}"] \\
(B \otimes C) \otimes A \arrow[r, "\alpha_{B,C,A}"'] & B \otimes (C \otimes A)
\end{tikzcd}$$

2.2 Smash Products

The model of the smash product we will use here is given by the cofibre of the inclusion $A \vee B \hookrightarrow A \wedge B$, i.e. the following pushout

$$\begin{tikzcd}
A \vee B \arrow[r] \arrow[d] & A \times B \arrow[d] \\
1 \arrow[r] & A \wedge B \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

For the sake of clarity, let us spell this out in detail:

Definition 4. *The smash product of two pointed types A and B, is the HIT generated by*

- *a point $\star_\wedge : A \wedge B$*
- *for every pair $(a, b) : A \times B$, a point $\langle a, b \rangle : A \wedge B$*
- *for every point $a : A$, a path $\mathsf{push}_\mathsf{l}(a) : \langle a, \star_B \rangle = \star_\wedge$*
- *for every point $b : B$, a path $\mathsf{push}_\mathsf{r}(b) : \langle \star_a, b \rangle = \star_\wedge$*
- *a coherence $\mathsf{push}_{\mathsf{lr}} : \mathsf{push}_\mathsf{l}(\star_A) = \mathsf{push}_\mathsf{r}(\star_B)$.*

We always take $A \wedge B$ to be pointed by $\star_\wedge$.

We remark we could equivalently have defined the smash product by the pushout

$$\begin{tikzcd}
A + B \arrow[r] \arrow[d] & A \times B \arrow[d] \\
1 + 1 \arrow[r] & A \wedge B \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

This definition has the advantage of not having any 2-dimensional path constructors but has the disadvantage of having additional an additional point constructor. It turns out that Definition 4 suits our purposes better, so we will stick with it.

Let us also verify that the smash product is functorial. In what follows, a pointed function $A \to_\star B$ is a function $f : A \to B$ equipped with a proof of pointedness $\star_f : f(\star_A) = \star_B$.

Definition 5. *For two pointed functions $f : A \to_\star C$ and $g : B \to_\star D$, there is an induced map $f \wedge g : A \wedge B \to_\star C \wedge D$ defined by*

$$\begin{aligned}
(f \wedge g)\,(\star_\wedge) &= \star_\wedge \\
(f \wedge g)\,\langle a, b \rangle &= \langle f(a), g(a) \rangle \\
\mathsf{ap}_{f \wedge g}(\mathsf{push}_\mathsf{l}(a)) &= \mathsf{ap}_{\langle f(a), - \rangle}(\star_g) \cdot \mathsf{push}_\mathsf{l}(f(a)) \\
\mathsf{ap}_{f \wedge g}(\mathsf{push}_\mathsf{r}(b)) &= \mathsf{ap}_{\langle -, g(b) \rangle}(\star_f) \cdot \mathsf{push}_\mathsf{r}(g(b)) \\
\mathsf{ap}_{\mathsf{ap}_{f \wedge g}}(\mathsf{push}_{\mathsf{lr}}) &= \ldots
\end{aligned}$$

where the omitted case is a simple coherence which will not matter for any future constructions or proofs. We take this map to be pointed by refl.

We also take the opportunity to mention the commutativity of smash products. This is trivial since the definition of $A \wedge B$ is entirely symmetric in both arguments.

Proposition 2. *The swap map $A \times B \to B \times A$ induces a pointed equivalence $A \wedge B \simeq_\star B \wedge A$.*

3 Associativity

Proving that the smash product is associative far less straightforward than proving its commutativity. In fact, even the seemingly direct task of constructing the associator map is no mean feat. While this has already been verified by van Dorn [Doo18] and, using a computer generated proof, by Brunerie [Bru18], let us give a direct construction of the equivalence. We do this because we will need it to be as easy to trace as possible when verifying e.g. the pentagon identity. For this purpose, we will introduce a new HIT capturing triple smash products in a way which is neutral with respect to the distribution of parentheses.

Definition 6. *Given pointed types A, B, C, we define the type $\bigwedge(A, B, C)$ as the HIT given by:*

- *a point* $\star_{3\wedge}$
- *for each triple of points* $(a, b, c) : A \times B \times C$*, a point* $\langle a, b, c\rangle : \bigwedge(A, B, C)$
- *for* $b : B$ *and* $c : C$*, a path* $\mathsf{push}_0(b, c) : \langle \star_A, b, c\rangle = \star_{3\wedge}$
- *for* $a : A$ *and* $c : C$*, a path* $\mathsf{push}_1(a, c) : \langle a, \star_B, c\rangle = \star_{3\wedge}$
- *for* $a : A$ *and* $b : B$*, a path* $\mathsf{push}_2(a, b) : \langle a, b, \star_C\rangle = \star_{3\wedge}$
- *for* $a : A$*, path* $\mathsf{push}_{1,2}(a) : \mathsf{push}_1(a, \star_C) = \mathsf{push}_2(a, \star_B)$
- *for* $b : B$*, path* $\mathsf{push}_{0,2}(b) : \mathsf{push}_0(b, \star_C) = \mathsf{push}_2(\star_A, b)$
- *for* $c : C$*, path* $\mathsf{push}_{0,1}(c) : \mathsf{push}_0(\star_A, c) = \mathsf{push}_1(\star_A, c)$
- *a coherence* $\mathsf{push}_{0,1,2}$ *filling*

$$\begin{tikzcd}
\mathsf{push}_0(\star_B, \star_C) \arrow[rr, "\mathsf{push}_{0,2}(\star_B)"] \arrow[dr, "\mathsf{push}_{0,1}(\star_C)"'] & & \mathsf{push}_2(\star_A, \star_B) \\
& \mathsf{push}_1(\star_A, \star_C) \arrow[ur, "\mathsf{push}_{1,2}(\star_A)"'] &
\end{tikzcd}$$

Let us verify that this actually captures a triple smash product. What we need is an equivalence $(A \wedge B) \wedge C \simeq \bigwedge(A, B, C)$. The equivalence is described in Table 1 with constructors of $(A \wedge B) \wedge C$ on the left and the corresponding constructors of $\bigwedge(A, B, C)$ on the right. We remark that this correspondence is only serves as an informal sketch of the equivalence—in practice, some simple coherences are needed for the higher constructors to make it well-typed. Verifying that this map indeed defines an equivalence is somewhat laborious but direct. We get the associativity of the smash product as a consequence.

Proposition 3. *There is a pointed equivalence* $\alpha_{A,B,C} : (A \wedge B) \wedge C \simeq_\star A \wedge (B \wedge C)$

Proof. We use the $\bigwedge(A, B, C)$ HIT to construct the equivalence and the fact that it is trivially invariant under permutation of the arguments in the sense that e.g. $\bigwedge(A, B, C) \simeq \bigwedge(C, A, B)$. We define $\alpha_{A,B,C}$ by the composite:

$$(A \wedge B) \wedge C \simeq \bigwedge(A, B, C) \simeq \bigwedge(C, B, A) \simeq (C \wedge B) \wedge A \simeq A \wedge (B \wedge C)$$

The fact that $\alpha_{A,B,C}$ is pointed holds by refl. □

$(A \wedge B) \wedge C$	$\to$	$\bigwedge(A, B, C)$
$\star_\wedge$	$\leadsto$	$\star_{3\wedge}$
$\langle \star_\wedge, c \rangle$	$\leadsto$	$\star_{3\wedge}$
$\langle\langle a, b\rangle, c\rangle$	$\leadsto$	$\langle a, b, c\rangle$
$\mathsf{ap}_{\langle -, c\rangle}(\mathsf{push}_\mathsf{l}(a))$	$\leadsto$	$\mathsf{push}_1(a, c)$
$\mathsf{ap}_{\langle -, c\rangle}(\mathsf{push}_\mathsf{r}(b))$	$\leadsto$	$\mathsf{push}_0(b, c)$
$\mathsf{ap}_{\mathsf{ap}_{\langle -, c\rangle}}(\mathsf{push}_\mathsf{lr})$	$\leadsto$	$\mathsf{push}_{0,1}(c)$
$\mathsf{push}_\mathsf{l}(\star_\wedge)$	$\leadsto$	refl
$\mathsf{push}_\mathsf{l}\langle a, b\rangle$	$\leadsto$	$\mathsf{push}_2(a, b)$
$\mathsf{ap}_{\mathsf{push}_\mathsf{l}}(\mathsf{push}_\mathsf{l}(a))$	$\leadsto$	$\mathsf{push}_{1,2}(a)$
$\mathsf{ap}_{\mathsf{push}_\mathsf{l}}(\mathsf{push}_\mathsf{r}(b))$	$\leadsto$	$\mathsf{push}_{0,2}(b)$
$\mathsf{ap}_{\mathsf{ap}_{\mathsf{push}_\mathsf{l}}}(\mathsf{push}_\mathsf{lr})$	$\leadsto$	$\mathsf{push}_{0,1,2}$
$\mathsf{push}_\mathsf{r}(c)$	$\leadsto$	refl
$\mathsf{push}_\mathsf{lr}$	$\leadsto$	refl

Table 1: $(A \wedge B) \wedge C$ vs. $\bigwedge(A, B, C)$

4 The Heuristic

Reasoning about functions defined over iterated smash products quickly gets out of hand. For instance, when verifying the pentagon axiom, we need to reason about functions on the form $((A \wedge B) \wedge C) \wedge D \to E$. To prove an equality of such functions f and g, we have to construct, for instance, a dependent path

$$\mathsf{ap}_{\mathsf{ap}_{\mathsf{ap}_f \circ \mathsf{push}_\mathsf{l}} \circ \mathsf{push}_\mathsf{l}}(\mathsf{push}_\mathsf{l}(a)) \;\rightsquigarrow\; \mathsf{ap}_{\mathsf{ap}_{\mathsf{ap}_g \circ \mathsf{push}_\mathsf{l}} \circ \mathsf{push}_\mathsf{l}}(\mathsf{push}_\mathsf{l}(a)) \tag{1}$$

which boils down to filling a 4-dimensional cube with highly non-trivial sides. This is often completely unmanageable in practice. The best thing we can hope for is that these types of coherence problems are, in some sense, automatic. This was, in fact, suggested in [Bru16], but never proved or in any way made formal. In this section, we will see that this, in fact, is the case.

The first troublesome part of verifying equalities of functions defined over smash products is the $\mathsf{push}_\mathsf{lr}$ constructor. Fortunately, we do not have to deal with it. Let us denote by $A \mathbin{\widetilde{\wedge}} B$ the smash product $A \wedge B$ minus the $\mathsf{push}_\mathsf{lr}$ constructor, i.e. the pushout

$$\begin{tikzcd}
A + B \arrow[r] \arrow[d] & A \times B \arrow[d] \\
1 \arrow[r] & A \mathbin{\widetilde{\wedge}} B \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

and let i be the inclusion $A \mathbin{\widetilde{\wedge}} B \hookrightarrow A \wedge B$.

Lemma 1. *For any two maps $f, g : A \wedge B \to C$ satisfying $f \circ i = g \circ i$, we have that $f = g$.*

Proof. The antecedent of the statement provides us with the following data:

- a path $p : f(\star_\wedge) = g(\star_\wedge)$
- a homotopy $h : ((a, b) : A \times B) \to f\langle a, b\rangle = g\langle a, b\rangle$

- for each $a : A$, a filler $h_l(a)$ of the square

$$\begin{tikzcd}
g\langle a, \star_B\rangle \arrow[r, "\mathsf{ap}_g(\mathsf{push}_\mathsf{l}(a))"] & g(\star_\wedge) \\
f\langle a, \star_B\rangle \arrow[u, "h(a,\star_B)"] \arrow[r, "\mathsf{ap}_f(\mathsf{push}_\mathsf{l}(a))"'] & f(\star_\wedge) \arrow[u, "p"']
\end{tikzcd}$$

- for each $b : B$, a filler $h_r(b)$ of the square

$$\begin{tikzcd}
g\langle \star_A, b\rangle \arrow[r, "\mathsf{ap}_g(\mathsf{push}_\mathsf{r}(b))"] & g(\star_\wedge) \\
f\langle \star_A, b\rangle \arrow[u, "h(\star_A,b)"] \arrow[r, "\mathsf{ap}_f(\mathsf{push}_\mathsf{r}(b))"'] & f(\star_\wedge) \arrow[u, "p"']
\end{tikzcd}$$

To prove that $f = g$, we need to provide a p', h', h'_l, h'_r of the same types as above, as well as a filler h'_{lr} of the cube

$$\begin{tikzcd}
g\langle \star_A, \star_B\rangle \arrow[rr, "\mathsf{ap}_g(\mathsf{push}_\mathsf{r}(\star_A))" description] \arrow[dr, equal] & & g(\star_\wedge) \arrow[dr, equal] & \\
& g\langle \star_A, \star_B\rangle \arrow[rr, "\mathsf{ap}_g(\mathsf{push}_\mathsf{l}(\star_A))" description] & & g(\star_\wedge) \\
f\langle \star_A, \star_B\rangle \arrow[uu] \arrow[rr, "\mathsf{ap}_f(\mathsf{push}_\mathsf{r}(\star_A))" description] \arrow[dr, equal] & & f(\star_\wedge) \arrow[uu] \arrow[dr, equal] & \\
& f\langle \star_A, \star_B\rangle \arrow[uu] \arrow[rr, "\mathsf{ap}_f(\mathsf{push}_\mathsf{l}(\star_A))" description] & & f(\star_\wedge) \arrow[uu]
\end{tikzcd}$$

where the top and bottom squares are given respectively by $\mathsf{ap}_{\mathsf{ap}_g}(\mathsf{push}_\mathsf{lr})$ and $\mathsf{ap}_{\mathsf{ap}_f}(\mathsf{push}_\mathsf{lr})$, the left-and right hand side respectively by $\mathsf{refl}_{h'(\star_A,\star_B)}$ and $\mathsf{refl}_{p'}$ and the front and back respectively by $h'_l(\star_A)$ and $h'_r(\star_B)$.

We set $p' = p$, $h' = h$ and $h'_l = h_l$. For h'_r, however, we need to make an alteration. We construct it as the following composite square

$$\begin{tikzcd}
g\langle \star_A, b\rangle \arrow[r, "\mathsf{ap}_g(\mathsf{push}_\mathsf{r}(b))"] & g(\star_\wedge) \arrow[r, equal] & g(\star_\wedge) \\
f\langle \star_A, b\rangle \arrow[u, "h(\star_A,b)"] \arrow[r, "\mathsf{ap}_f(\mathsf{push}_\mathsf{r}(b))"'] & f(\star_\wedge) \arrow[u, "p"'] \arrow[r, equal] & f(\star_\wedge) \arrow[u, "p"']
\end{tikzcd}$$

where the square on the left is $h_r(b)$ and the square on the right is the lid of the following cube

$$\begin{tikzcd}
g(\star_\wedge) \arrow[rr, equal] & & g(\star_\wedge) & \\
& f(\star_\wedge) \arrow[ul, "p"'] \arrow[rr, equal] & & f(\star_\wedge) \arrow[ul, "p"'] \\
g\langle \star_A, \star_B \rangle \arrow[uu, "\mathsf{ap}_g(\mathsf{push_l}(\star_A))"] \arrow[rr, equal] & & g\langle \star_A, \star_B \rangle \arrow[uu, "\mathsf{ap}_g(\mathsf{push_r}(\star_B))"] & \\
& f\langle \star_A, \star_B \rangle \arrow[ul, "h(\star_A,\star_B)"] \arrow[uu, "\mathsf{ap}_f(\mathsf{push_l}(\star_A))"] \arrow[rr, equal] & & f\langle \star_A, \star_B \rangle \arrow[ul, "h(\star_A,\star_B)"] \arrow[uu, "\mathsf{ap}_f(\mathsf{push_r}(\star_B))"]
\end{tikzcd}$$

whose sides are given by $h_l(\star_A)$ and $h_r(\star_B)$ on the left and right respectively, the action of f and g on $\mathsf{push_{lr}}$ on the front and back respectively and $\mathsf{refl}_{h(\star_A,\star_B)}$ on the bottom. One can now easily construct the filler h'_{lr} by generalising the cubes involved and applying path induction. □

This lemma is very useful but does not get us all the way. Complicated paths like in (1) still need to be constructed, regardless of what happens with the $\mathsf{push_{lr}}$ constructor. To strengthen the principle, we will need to introduce *homogeneous types*.

Definition 7. *A pointed type A is homogeneous if for any $a : A$ there is a pointed equivalence $(A, \star_A) \simeq_\star (A, a)$.*

The usefulness of homogeneous types is showcased in the following lemma which was first conjectured for Eilenberg-MacLane spaces in work leading up to [BLM22] and later proved and generalised Cavallo (and later further generalised by Buchholtz, Christensen, G. Taxerås Flaten and Rijke [Buc+23]).

Lemma 2. *Let $f, g : A \to_\star B$ with B homogeneous. If $f = g$ as plain functions, then $f = g$ as pointed functions.*

The same lemma holds for bi-pointed functions $f, g : A \to_\star (B \to_\star C)$, since the type $(B \to_\star C)$ is homogeneous if C is. This gives a corresponding principle for maps defined over smash products via the adjunction

$$(A \wedge B \to_\star C) \simeq (A \to_\star (B \to_\star C))$$

Lemma 3. *Let $f, g : A \wedge B \to_\star C$ with C homogeneous. If $f\langle a, b\rangle = g\langle a, b\rangle$ for all $a : A$ and $b : B$, we get an equality of pointed functions $f = g$.*

If we could apply Lemma 3 when proving the pentagon identity, we would be done immediately. Unfortunately, none of the types showing up in the pentagon is homogeneous. There is, however, some use for it. Let us first make the following observation: given two pointed functions $f, g : A \wedge B \to_\star C$ and a homotopy $h : ((a, b) : A \times B) \to f\langle a, b\rangle = g\langle a, b\rangle$, we can define two functions $L_h : A \to \Omega C$ and $R_h : B \to \Omega C$ in terms of h (and suitable coherences). The obvious definition of these maps will give us a version of Lemma 1 which tells us that if L_h and R_h are constant, then $f = g$ as plain functions and, furthermore, that if L_h or R_h is pointed, then $f = g$ as pointed functions. We can weaken this by requiring that L_h and R_h both are equal, as pointed functions, to the constant functions sending any point to refl. If either A or B is another smash product, this *would* be a situation where Lemma 3 applies since ΩC (and indeed any path type) is homogeneous.

Let us spell this out. In fact, we can state this idea without any pointedness assumptions.

Definition 8. *Let $f, g : A \wedge B \to C$ and let $h : ((a, b) : A \times B) \to f\langle a, b\rangle = g\langle a, b\rangle$. This induces, in particular, functions $L_h : A \to_\star f(\star_\wedge) = g(\star_\wedge)$ and $R_h : B \to_\star f(\star_\wedge) = g(\star_\wedge)$ defined by*

$$\begin{aligned} L_h(a) &= (\mathsf{ap}_f(\mathsf{push}_\mathsf{l}(a)))^{-1} \cdot h(a, \star_B) \cdot \mathsf{ap}_g(\mathsf{push}_\mathsf{l}(b)) \\ R_h(b) &= (\mathsf{ap}_f(\mathsf{push}_\mathsf{r}(b)))^{-1} \cdot h(\star_A, b) \cdot \mathsf{ap}_g(\mathsf{push}_\mathsf{r}(b)) \end{aligned}$$

where we may simply take $f(\star_\wedge) = g(\star_\wedge)$ to be pointed by either $L_h(\star_A)$ or $R_h(\star_A)$ (these are equal by $\mathsf{push}_\mathsf{lr}$, so the choice does not matter).

We may use L_h and R_h to give a compact induction principle for identities $f = g$. The following lemma is a direct weakening of Lemma 1 along the lines of the previous discussion.

Lemma 4. *Let $f, g : A \wedge B \to C$. The following data gives an equality $f = g$:*

- *A homotopy $h : ((a, b) : A \times B) \to f\langle a, b\rangle = g\langle a, b\rangle$*
- *Equalities of pointed functions $L_h = \mathsf{const}_{L_h(\star_B)}$ and $R_h = \mathsf{const}_{R_r(\star_B)}$.*
- *(Optional, if an equality of pointed functions is desired): A coherence $\star_f = L_h(\star_B) \cdot \star_g$*

Let us stress the key idea again, since one very reasonably may ask why Lemma 4 is useful in any way—asking for an equality of pointed functions in the second datum when, in fact, only an equality of regular functions is needed seems unnecessary—even if the codomains of L_h and R_h are homogeneous and pointed equality is logically equivalent to regular equality by Lemma 2. There is, however, a point to pointedness here. When either A or B (or both) is another smash product, the fact that L_h and R_h have homogeneous codomain means that Lemma 3 applies. Let us exemplify this:

Lemma 5. *For any two functions $f, g : (A \wedge B) \wedge C \to D$, the following data gives an equality $f = g$:*

(i) A homotopy $h : ((a, b, c) : A \times B \times C) \to f\langle a, b, c\rangle = g\langle a, b, c\rangle$.

(ii) For every pair $(a, b) : A \times B$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, \star_C\rangle \arrow[r, "h(\star_A,\star_B,\star_C)"] & g\langle \star_A, \star_B, \star_C\rangle \\
f\langle \star_\wedge, \star_C\rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B)^{-1}))"] & g\langle \star_\wedge, \star_C\rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B)^{-1}))"'] \\
f(\star_\wedge) \arrow[u, "\mathsf{ap}_f(\mathsf{push}_\mathsf{l}(\star_\wedge)^{-1})"] & g(\star_\wedge) \arrow[u, "\mathsf{ap}_g(\mathsf{push}_\mathsf{l}(\star_\wedge)^{-1})"'] \\
f\langle a, b, \star_C\rangle \arrow[u, "\mathsf{ap}_f(\mathsf{push}_\mathsf{l}\langle a,b\rangle)"] \arrow[r, "h(a,b,\star_C)"'] & g\langle a, b, \star_C\rangle \arrow[u, "\mathsf{ap}_g(\mathsf{push}_\mathsf{l}\langle a,b\rangle)"']
\end{tikzcd}$$

(iii) For every point $c : C$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, \star_C\rangle \arrow[r, "h(\star_A,\star_B,\star_C)"] & g\langle \star_A, \star_B, \star_C\rangle \\
f\langle \star_\wedge, \star_C\rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}"] & g\langle \star_\wedge, \star_C\rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}"'] \\
f(\star_\wedge) \arrow[u, "\mathsf{ap}_f(\mathsf{push}_\mathsf{l}(\star_\wedge))^{-1}"] & g(\star_\wedge) \arrow[u, "\mathsf{ap}_g(\mathsf{push}_\mathsf{l}(\star_\wedge))^{-1}"'] \\
f\langle \star_\wedge, c\rangle \arrow[u, "\mathsf{ap}_f(\mathsf{push}_\mathsf{r}(c))"] & g\langle \star_\wedge, c\rangle \arrow[u, "\mathsf{ap}_g(\mathsf{push}_\mathsf{r}(c))"'] \\
f\langle \star_A, \star_B, c\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))"] \arrow[r, "h(\star_A,\star_B,c)"'] & g\langle \star_A, \star_B, c\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))"']
\end{tikzcd}$$

(iv) For every pair $(a, c) : A \times C$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, c\rangle \arrow[r, "h(\star_A,\star_B,c)"] & g\langle \star_A, \star_B, c\rangle \\
f\langle \star_\wedge, c\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c\rangle}(\mathsf{push}_\mathsf{l}(\star_A))^{-1}"] & g\langle \star_\wedge, c\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c\rangle}(\mathsf{push}_\mathsf{l}(\star_A))^{-1}"'] \\
f\langle a, \star_B, c\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c\rangle}(\mathsf{push}_\mathsf{l}(a))"] \arrow[r, "h(a,\star_B,c)"'] & g\langle a, \star_B, c\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c\rangle}(\mathsf{push}_\mathsf{l}(a))"']
\end{tikzcd}$$

(v) For every pair $(b, c) : B \times C$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, c\rangle \arrow[r, "h(\star_A,\star_B,c)"] & g\langle \star_A, \star_B, c\rangle \\
f\langle \star_\wedge, c\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}"] & g\langle \star_\wedge, c\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}"'] \\
f\langle \star_A, b, c\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(b))"] \arrow[r, "h(\star_A,b,c)"'] & g\langle \star_A, b, c\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(b))"']
\end{tikzcd}$$

Proof. Suppose we have the given data. We apply Lemma 4 to f and g. This breaks the proof up in 2 subgoals:

- First, we need to provide a homotopy $k : ((x, c) : (A \wedge B) \times C) \to f\langle x, c\rangle = g\langle x, c\rangle$. To construct k, we may apply Lemma 4 (or, equivalently in this case since we only have one copy of the smash product in the domain, Lemma 1). This gives us 2 new subgoals.
  - First, we need a homotopy $((a, b, c) : A \times B \times C) \to f\langle a, b, c\rangle = g\langle a, b, c\rangle$. This is given by h.
  - We need to show that $L_{h(-,c)}$ and $R_{h(-,c)}$ are constant. Since their codomains are homogeneous, it provides to prove the equality for underlying functions. This boils down to providing fillers of the squares which we assumed in (iv) and (v).

- We then need to show that L_k and R_k are constant. To show that L_k is constant, we apply Lemma 3 using that its codomain is homogeneous. Hence, we only need to verify that $L_k\langle a, b\rangle = L_k(\star_\wedge)$. Unfolding the definitions, we see that this is given to by assumption (ii). Note that this is where the explosion of complexity would normally happen but, thanks to Lemma 3, we completely avoid having to verify any higher coherences. For R_k, it suffices by Lemma 2 to show that its underlying function is constant. Again, unfolding the definitions, we see that this is given by (iii).

□

For completeness, let us state the corresponding lemma for functions $f, g : ((A\wedge B)\wedge C)\wedge D) \to E$. The proof is by Lemma 4 and Lemma 5, following the exact same line of attack as the proof of Lemma 5. We stress that it is not important to read the statement in detail since it is rather technical—we mainly include it to showcase the fact that only squares are involved as opposed to the (many) high-dimensional cubes of coherences which would appear in a naïve inductive proof.

Lemma 6. *For any two functions $f, g : ((A \wedge B) \wedge C) \wedge D) \to E$, the following data gives an equality $f = g$:*

(i) A homotopy $h : ((a, b, c, d) : A \times B \times C \times D) \to f\langle a, b, c, d\rangle = g\langle a, b, c, d\rangle$.

(ii) For every triple $(a, b, d) : A \times B \times D$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, \star_C, d\rangle \arrow[r, "h(\star_A,\star_B,\star_C)"] & g\langle \star_A, \star_B, \star_C, d\rangle \\
f\langle \star_\wedge, \star_C, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B)^{-1}))"] & g\langle \star_\wedge, \star_C, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B)^{-1}))"'] \\
f\langle \star_\wedge, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)^{-1})"] & g\langle \star_\wedge, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)^{-1})"'] \\
f\langle a, b, \star_C, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}\langle a,b\rangle)"] \arrow[r, "h(a,b,\star_C)"'] & g\langle a, b, \star_C, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}\langle a,b\rangle)"']
\end{tikzcd}$$

(iii) For every pair $(c, d) : C \times D$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, \star_C, d\rangle \arrow[r, "h(\star_A,\star_B,\star_C,d)"] & g\langle \star_A, \star_B, \star_C\rangle \\
f\langle \star_\wedge, \star_C, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B))^{-1}"] & g\langle \star_\wedge, \star_C\rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B))^{-1}"'] \\
f\langle \star_\wedge, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge))^{-1}"] & g\langle \star_\wedge, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge))^{-1}"'] \\
f\langle \star_\wedge, c, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_{\mathsf{r}}(c))"] & g\langle \star_\wedge, c, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_{\mathsf{r}}(c))"'] \\
f\langle \star_A, \star_B, c, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B))"] \arrow[r, "h(\star_A,\star_B,c,d)"'] & g\langle \star_A, \star_B, c, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B))"']
\end{tikzcd}$$

(iv) For every triple $(a, c, d) : A \times C \times D$*, a filler of the square*

$$\begin{tikzcd}
f\langle \star_A, \star_B, c, d\rangle \arrow[r, "h(\star_A,\star_B,c,d)"] & g\langle \star_A, \star_B, c, d\rangle \\
f\langle \star_\wedge, c, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A))^{-1}"] & g\langle \star_\wedge, c, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A))^{-1}"'] \\
f\langle a, \star_B, c, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(a))"] \arrow[r, "h(a,\star_B,c,d)"'] & g\langle a, \star_B, c, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(a))"']
\end{tikzcd}$$

(v) For every triple $(b, c, d) : B \times C \times D$*, a filler of the square*

$$\begin{tikzcd}
f\langle \star_A, \star_B, c, d\rangle \arrow[r, "h(\star_A,\star_B,c,d)"] & g\langle \star_A, \star_B, c, d\rangle \\
f\langle \star_\wedge, c, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B))^{-1}"] & g\langle \star_\wedge, c, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B))^{-1}"'] \\
f\langle \star_A, b, c, d\rangle \arrow[u, "\mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{r}}(b))"] \arrow[r, "h(\star_A,b,c,d)"'] & g\langle \star_A, b, c, d\rangle \arrow[u, "\mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{r}}(b))"']
\end{tikzcd}$$

(vi) For every triple $(a, b, c) : A \times B \times C$*, a filler of the square*

$$\begin{tikzcd}
f\langle \star_A, \star_B, \star_C\star_D\rangle \arrow[r, "h(\star_A,\star_B,\star_C\star_D)"] & g\langle \star_A, \star_B, \star_C\star_D\rangle \\
f\langle \star_\wedge, \star_C, \star_D\rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A)))^{-1}"] & g\langle \star_\wedge, \star_C, \star_D\rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A)))^{-1}"'] \\
f\langle \star_\wedge, \star_D\rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}"] & g\langle \star_\wedge, \star_D\rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}"'] \\
f(\star_\wedge) \arrow[u, "\mathsf{ap}_{f}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}"] & g(\star_\wedge) \arrow[u, "\mathsf{ap}_{g}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}"'] \\
f\langle a, b, c, \star_D\rangle \arrow[u, "\mathsf{ap}_{f}(\mathsf{push}_{\mathsf{l}}\langle a,b,c\rangle)"] \arrow[r, "h(a,b,c,\star_D)"'] & g\langle a, b, c, \star_D\rangle \arrow[u, "\mathsf{ap}_{g}(\mathsf{push}_{\mathsf{l}}\langle a,b,c\rangle)"']
\end{tikzcd}$$

(vii) For every $d : D$, a filler of the square

$$\begin{tikzcd}
f\langle \star_A, \star_B, \star_C, \star_D \rangle \arrow[r, "h(\star_A,\star_B,\star_C,\star_D)"] & g\langle \star_A, \star_B, \star_C, \star_D \rangle \\
f\langle \star_\wedge, \star_C, \star_D \rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C,\star_D\rangle}(\mathsf{push}_\mathsf{l}(\star_A)))^{-1}"] & g\langle \star_\wedge, \star_C, \star_D \rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C,\star_D\rangle}(\mathsf{push}_\mathsf{l}(\star_A)))^{-1}"'] \\
f\langle \star_\wedge, \star_D \rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_D\rangle}(\mathsf{push}_\mathsf{l}(\star_\wedge)))^{-1}"] & g\langle \star_\wedge, \star_D \rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_D\rangle}(\mathsf{push}_\mathsf{l}(\star_\wedge)))^{-1}"'] \\
f(\star_\wedge) \arrow[u, "\mathsf{ap}_{f}(\mathsf{push}_\mathsf{l}(\star_\wedge)))^{-1}"] & g(\star_\wedge) \arrow[u, "\mathsf{ap}_{g}(\mathsf{push}_\mathsf{l}(\star_\wedge)))^{-1}"'] \\
f\langle \star_\wedge, d \rangle \arrow[u, "\mathsf{ap}_{f}(\mathsf{push}_\mathsf{r}(d))"] & g\langle \star_\wedge, d \rangle \arrow[u, "\mathsf{ap}_{g}(\mathsf{push}_\mathsf{r}(d))"'] \\
f\langle \star_\wedge, \star_C, d \rangle \arrow[u, "\mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_\mathsf{l}(\star_\wedge)))"] & g\langle \star_\wedge, \star_C, d \rangle \arrow[u, "\mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_\mathsf{l}(\star_\wedge)))"'] \\
f\langle \star_A, \star_B, \star_C, d \rangle \arrow[u, "\mathsf{ap}_{f\langle -,\star_C,d\rangle}(\mathsf{push}_\mathsf{l}(\star_A)))"] \arrow[r, "h(\star_A,\star_B,\star_C,d)"'] & g\langle \star_A, \star_B, \star_C, d \rangle \arrow[u, "\mathsf{ap}_{g\langle -,\star_C,d\rangle}(\mathsf{push}_\mathsf{l}(\star_A)))"']
\end{tikzcd}$$

Let us make three observations about Lemma 5 and Lemma 6.

1. In both statements, the different pieces of data are almost completely mutually independent. The only choice we make that matters is the choice of homotopy h. This means that we are free to provide any proofs we like for the remaining steps without having to worry about future coherences.

2. In many cases (especially those relating to the symmetric monoidal structure of the smash product) the homotopy h will be defined by $h(a_1, \dots, a_n) = \mathsf{refl}$. This means that all other data we need to provide are equalities of composite paths. In addition, all paths involved are defined in terms of applications of f and g on path constructors which we can usually simply unfold to something (hopefully) simple.

3. Going from Lemma 5 to Lemma 6, we see that only two additional assumptions are needed. If we would to increase the number of copies of smash products appearing in the domain by one, we would only need to provide two additional squares (and still no higher coherences). Hence, the complexity of such proofs grows linearly with the complexity of the domain. For comparison, if we were to resort to a naïve proof by deep smash product induction, the amount of data needed would grow exponentially.

While it is rather difficult to provide a generalisation of Lemma 5 and Lemma 6 for arbitrarily large iterations of the smash product in a way that both informative and is useful in practice, we can at least state the general idea as an informal heuristic:

Heuristic. *To show that two functions $f, g : \bigwedge_{i \leq n} A_i \to B$ are equal, it suffices, by iterative application Lemmas 2 to 4, to provide a family of paths $h(x_1, \dots, x_n) : f\langle x_1, \dots, x_n \rangle = g\langle x_1, \dots, x_n \rangle$ for $x_i : A_i$ and to show that this is coherent with f and g on any* single *application of* push_l *or*

$\mathsf{push_r}$ *(e.g.* $\mathsf{ap}_{\langle -,x_{i+1},\ldots,x_n\rangle}(\mathsf{push_l}\langle x_1,\ldots,x_{i-1}\rangle)$*). Furthermore, if an equality of pointed functions is required, we need to provide a filler of the following square:*

$$\begin{tikzcd}
f\langle \star_{A_1},\ldots,\star_{A_n}\rangle \arrow[rr, "h(\star_{A_1},\ldots,\star_{A_n})"] \arrow[d] & & g\langle \star_{A_1},\ldots,\star_{A_n}\rangle \arrow[d] \\
\vdots \arrow[d] & & \vdots \arrow[d] \\
f\langle \star_\wedge,\star_{A_n}\rangle \arrow[d, swap, "\mathsf{ap}_f(\mathsf{push_r}(\star_A))"] & & g\langle \star_\wedge,\star_{A_n}\rangle \arrow[d, "\mathsf{ap}_g(\mathsf{push_r}(\star_A))"] \\
f(\star_\wedge) \arrow[r, swap, "\star_f"] & \star_B \arrow[r, swap, "\star_g^{-1}"] & g(\star_\wedge)
\end{tikzcd}$$

5 The Symmetric Monoidal Structure

Let us reap the fruits of our labour and show that $\wedge$ defines a symmetric monoidal product on the universe of pointed types. We will not verify all axioms here, since this is neither very instructive nor very interesting. Instead, we sketch the proofs of the most technical properties.

Proposition 4. *The smash product satisfies the hexagon axiom, i.e. we have an equality of pointed functions $H_0 = H_1$ where H_0 and H_1 are defined as the composites of each side of the pentagon:*

$$\begin{tikzcd}
(A\wedge B)\wedge C \arrow[r, "\beta_{A,B}\wedge 1_C"] \arrow[d, swap, "\alpha_{A,B,C}"] \arrow[drr, dashed, "H_1"] \arrow[dr, dashed, swap, "H_0"] & (B\wedge A)\wedge C \arrow[d, "\alpha_{B,A,C}"] & \\
A\wedge(B\wedge C) \arrow[d, swap, "\beta_{A,B\wedge C}"] & B\wedge(A\wedge C) \arrow[d, "1_B\wedge\beta_{A,C}"] & \\
(B\wedge C)\wedge A \arrow[r, swap, "\alpha_{B,C,A}"] & B\wedge(C\wedge A) &
\end{tikzcd}$$

Proof. We show the statement by an application of our heuristic which, in this case, takes the form of Lemma 5. We provide the data as follows:

1. For the homotopy $h(a,b,c) : H_0\langle a,b,c\rangle = H_1\langle a,b,c\rangle$ we simple choose $h(a,b,c) = \mathsf{refl}$, since both sides compute to $\langle b,c,a\rangle$.

2. The second datum in Lemma 5 computes to[1] the following square filling problem:

$$\begin{tikzcd}
\langle \star_B, \star_C, \star_A \rangle \arrow[r, "\mathsf{refl}"] & \langle \star_B, \star_C, \star_A \rangle \\
\wedge_C \arrow[u, "\mathsf{push_l}(\star_B)^{-1} \cdot \mathsf{ap}_{\langle \star_B, - \rangle}(\mathsf{push_l}(\star_C))^{-1}"] & \wedge_C \arrow[u, "\mathsf{push_l}(\star_B)^{-1} \cdot \mathsf{ap}_{\langle \star_B, - \rangle}(\mathsf{push_l}(\star_C))^{-1}"'] \\
\wedge_C \arrow[u, "\mathsf{refl}"] & \wedge_C \arrow[u, "\mathsf{refl}"'] \\
\langle b, \star_C, a \rangle \arrow[u, "\mathsf{ap}_{\langle b, - \rangle}(\mathsf{push_r}(a)) \cdot \mathsf{push_l}(b)"] \arrow[r, "\mathsf{refl}"'] & \langle b, \star_C, a \rangle \arrow[u, "\mathsf{ap}_{\langle b, - \rangle}(\mathsf{push_r}(a)) \cdot \mathsf{push_l}(b)"']
\end{tikzcd}$$

which of is solved by refl.

3. The remaining squares are computed and solved in an identical manner.

4. For the pointedness, we need to fill the square outlined in end of the statement of the heuristic. This is equally direct since $\star_{H_0} = \star_{H_1} = \mathsf{refl}$, which holds because all functions involved in the definitions of H_0 and H_1 are pointed by refl.

□

Proposition 5. *The pentagon identity holds for the smash product, i.e. we have an equality of pointed functions $P_0 = P_1$ where P_0 and P_1 are defined as the composites of each side of the pentagon:*

$$\begin{tikzcd}
& ((A \wedge B) \wedge C) \wedge D \arrow[dl, "\alpha_{A,B,C} \wedge 1_D"'] \arrow[dr, "\alpha_{A \wedge B, C, D}"] \arrow[ddr, dashed, bend right, "P_0"] \arrow[ddr, dashed, "P_1"] & \\
(A \wedge (B \wedge C)) \wedge D \arrow[d, "\alpha_{A, B \wedge C, D}"'] & & (A \wedge B) \wedge (C \wedge D) \arrow[d, "\alpha_{A,B,C \wedge D}"] \\
A \wedge ((B \wedge C) \wedge D) \arrow[rr, "1_A \wedge \alpha_{B,C,D}"'] & & A \wedge (B \wedge (C \wedge D))
\end{tikzcd}$$

Proof. The statement follows easily by the heuristic, which in this case corresponds to Lemma 6. The proof is identical to the proof of Proposition 4 and follows simply by evaluating P_0 and P_1 on the 1-dimensional path constructors involved and noting that all square-filling problems listed in Lemma 6 become trivial. □

All other axioms defining a symmetric monoidal wild categories follow in the same direct manner and we, after some rather mechanical labour, easily arrive at the main result.

Theorem 1. *The universe of pointed types forms a symmetric monoidal wild category with the smash product as tensor product.*

[1] By 'computes to' we do not mean 'normalises in `Agda` to'. We mean 'compute' in the manual sense, i.e. by tracing H_0 and H_1 on the point and path constructors involved. Direct normalisation in `Agda` produces rather large and unmanageable terms. However, using `Agda` to normalise the terms in a more controlled manner (i.e. step-by-step) is very useful, as a sanity check, for inspecting the action of H_0 and H_1 on the path constructors involved.

References

[Bru18] G. Brunerie. "Computer-generated proofs for the monoidal structure of the smash product". *Homotopy Type Theory Electronic Seminar Talks*. Nov. 2018. URL: https://www.uwo.ca/math/faculty/kapulkin/seminars/hottest.html.

[Bru16] G. Brunerie. "On the homotopy groups of spheres in homotopy type theory". PhD thesis. Université Nice Sophia Antipolis, 2016. URL: http://arxiv.org/abs/1606.05916.

[BLM22] G. Brunerie, A. Ljungström, and A. Mörtberg. "Synthetic Integral Cohomology in Cubical Agda". In: *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*. Ed. by F. Manea and A. Simpson. Vol. 216. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 11:1–11:19. ISBN: 978-3-95977-218-1. DOI: 10.4230/LIPIcs.CSL.2022.11. URL: https://drops.dagstuhl.de/opus/volltexte/2022/15731.

[Buc+23] U. Buchholtz et al. "Central H-spaces and banded types" (2023). DOI: 10.48550/ARXIV.2301.02636. URL: https://arxiv.org/abs/2301.02636.

[CK17] P. Capriotti and N. Kraus. "Univalent Higher Categories via Complete Semi-Segal Types". *Proceedings of the ACM on Programming Languages* 2.POPL (Dec. 2017), 44:1–44:29. ISSN: 2475-1421. DOI: 10.1145/3158132.

[Cav21] E. Cavallo. "Higher Inductive Types and Internal Parametricity for Cubical Type Theory". PhD thesis. Carnegie Mellon University, Feb. 2021. URL: https://kilthub.cmu.edu/articles/thesis/Higher_Inductive_Types_and_Internal_Parametricity_for_Cubical_Type_Theory/14555691.

[CH20] E. Cavallo and R. Harper. "Internal Parametricity for Cubical Type Theory". In: *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*. Ed. by M. Fernández and A. Muscholl. Vol. 152. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, 13:1–13:17. ISBN: 978-3-95977-132-0. DOI: 10.4230/LIPIcs.CSL.2020.13.

[Doo18] F. van Doorn. "On the Formalization of Higher Inductive Types and Synthetic Homotopy Theory". PhD thesis. Carnegie Mellon University, May 2018. URL: https://arxiv.org/abs/1808.10690.

[LM23] A. Ljungström and A. Mörtberg. *Formalizing $\pi_4(\mathbb{S}^3) \cong \mathbb{Z}/2\mathbb{Z}$ and Computing a Brunerie Number in Cubical Agda*. 2023. DOI: 10.48550/ARXIV.2302.00151. URL: https://arxiv.org/abs/2302.00151.

[UF13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: Self-published, 2013. URL: https://homotopytypetheory.org/book/.