Mathematical Structures in Computer Science (2024), **34**, pp. 985–1007
doi:10.1017/S0960129524000318

CAMBRIDGE UNIVERSITY PRESS

PAPER

Symmetric monoidal smash products in homotopy type theory

Axel Ljungström

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

(Received 31 January 2024; revised 7 August 2024; accepted 4 September 2024; first published online 4 November 2024)

Abstract

In homotopy type theory, few constructions have proved as troublesome as the smash product. While its definition is just as direct as in classical mathematics, one quickly realises that in order to define and reason about functions over iterations of it, one has to verify an exponentially growing number of coherences. This has led to crucial results concerning smash products remaining open. One particularly important such result is the fact that the smash product forms a (1-coherent) symmetric monoidal product on the universe of pointed types. This fact was used, without a complete proof, by, for example, Brunerie ((2016) PhD thesis, Université Nice Sophia Antipolis) to construct the cup product on integral cohomology and is, more generally, a fundamental result in traditional algebraic topology. In this paper, we salvage the situation by introducing a simple informal heuristic for reasoning about functions defined over iterated smash products. We then use the heuristic to verify, for example, the hexagon and pentagon identities, thereby obtaining a proof of symmetric monoidality. We also provide a formal statement of the heuristic in terms of an induction principle concerning the construction of homotopies of functions defined over iterated smash products. The key results presented here have been formalised in the proof assistant Cubical Agda.

Keywords: Smash products; synthetic homotopy theory; symmetric monoidal categories; homotopy type theory; univalent foundations; constructive mathematics

1. Introduction

In his 2016 proof of $\pi_4(S^3) \cong \mathbb{Z}/2\mathbb{Z}$ in homotopy type theory (HoTT), Brunerie (2016) crucially uses – but never proves in detail – that the smash product is (1-coherent) symmetric monoidal. Due to the vast amount of path algebra involved when reasoning about smash products in HoTT, this has since remained open. While it turns out that smash products are not needed for Brunerie's proof (Ljungström and Mörtberg 2023), the problem is still interesting in its own right.

Several attempts have been made at salvaging the situation. van Doorn (2018) came very close by considering an argument using closed monoidal categories but left a gap where the path algebra became too technical. To be more precise, Van Doorn never verified that the equivalence

$$(A \wedge B \to_\star C) \simeq (A \to_\star (B \to_\star C)) \tag{1}$$

is a pointed natural equivalence. Another line of attack by Cavallo and Harper (Cavallo and Harper 2020; Cavallo 2021a) 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 expense of making the type theory

more complicated. Yet another solution was studied by Brunerie (2018) who used Agda metaprogramming 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 high memory consumption.

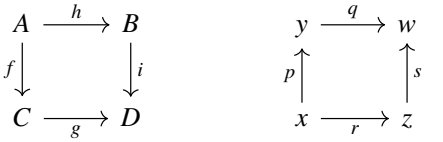
In this paper, we provide another solution by introducing an informal heuristic for reasoning about functions defined over iterated smash products. This approach vastly reduces the complexity of the identity proofs involved. We use this to give a complete proof of the fact that the smash product is 1-coherent symmetric monoidal. We finally discuss how to make the heuristic formal and express (a version of it) as a theorem which we also prove (see Theorem 30).

The paper is structured as follows. In Section 2, we introduce the two fundamental concepts of interest, namely symmetric monoidal wild categories and smash products. Section 3 introduces a new higher inductive type (HIT) capturing double smash products. We use this to sketch the construction of the associator map $\alpha_{A,B,C} : (A \wedge B) \wedge C \xrightarrow{\sim} A \wedge (B \wedge C)$. In Section 4, we make a small detour and discuss induction principles for the smash product. It is here that we introduce the heuristic mentioned above. In Section 5, we apply our heuristic and sketch the proof of the symmetric monoidality of the smash product. Finally, in Section 6, we discuss the translation of our heuristic into a formal result. We provide one suggestion and prove it correct.

This paper is written in the informal flavour of type theory employed in, for example, The Univalent Foundations Program (2013). Nevertheless, all key results have been formalised in the proof assistant Cubical Agda (Vezzosi et al. 2021), a cubical extension of Agda which, in particular, enjoys native support for HITs. A file summarising the relevant formalisations can be found in the agda/cubical library at https://github.com/agda/cubical/blob/master/Cubical/Papers/SmashProducts.agda (and, alternatively, on a frozen branch at https://github.com/aljungstrom/cubical/blob/smashpaper/Cubical/Papers/SmashProducts.agda).

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 (The Univalent Foundations Program 2013) for the basic constructions and definitions used here. For the remainder of this paper, we adopt the convention that square-shaped diagrams denote (i) commutative squares of functions whenever their source is in the top-left corner and (ii) commutative squares of paths whenever their source is in the bottom-left corner. For instance, the left diagram below expresses the identity (of functions) $g \circ f = i \circ h$, whereas the right diagram below expresses the identity (of paths) $p \cdot q = r \cdot s$.



2.1 Symmetric monoidal wild categories

To make the statements in this paper somewhat more compact, we employ the framework of *wild categories*. These are defined similarly to categories but without the condition that the morphisms form a set (Capriotti and Kraus 2017):

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

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

The difference between a wild category and a category is that the latter asks for *sets* of morphisms. In this paper, the wild category of interest is that of pointed types (at some universe level which is left implicit in this paper).

Proposition 2. *Let* $\mathsf{Type}_\star$ *denote the universe of pointed types (at some universe level). This universe forms a wild category with* $\mathsf{Type}_\star[A, B] :\equiv (A \to_\star B)$*, that is, 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 also a *symmetric monoidal* wild category, so let us define this.

Definition 3 (Monoidal wild categories). *A monoidal wild category is a wild category M with*

- *a functor* $\otimes : M \times M \to M$,
- *a unit, that is, an element* $I : M$ *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 identity holds, that is, the following diagram commutes,*

$$\begin{array}{ccc} (A \otimes I) \otimes B & \xrightarrow{\alpha_{A,I,B}} & A \otimes (I \otimes B) \\ & {\scriptstyle \rho_A \otimes 1_B} \searrow \quad \swarrow {\scriptstyle 1_A \otimes \lambda_B} & \\ & A \otimes B & \end{array}$$

  - *the pentagon identity holds, that is, the following diagram commutes.*

$$\begin{array}{ccc} & ((A \otimes B) \otimes C) \otimes D & \\ {\scriptstyle \alpha_{A,B,C} \otimes 1_D} \swarrow & & \searrow {\scriptstyle \alpha_{A \otimes B,C,D}} \\ (A \otimes (B \otimes C)) \otimes D & & (A \otimes B) \otimes (C \otimes D) \\ {\scriptstyle \alpha_{A,B \otimes C,D}} \downarrow & & \downarrow {\scriptstyle \alpha_{A,B,C \otimes D}} \\ A \otimes ((B \otimes C) \otimes D) & \xrightarrow{1_A \otimes \alpha_{B,C,D}} & A \otimes (B \otimes (C \otimes D)) \end{array}$$

Definition 4 (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, that is, the following diagram commutes.*

$$\begin{array}{ccc} (A \otimes B) \otimes C & \xrightarrow{\beta_{A,B} \otimes 1_C} & (B \otimes A) \otimes C \\ {\scriptstyle \alpha_{A,B,C}} \downarrow & & \downarrow {\scriptstyle \alpha_{B,A,C}} \\ A \otimes (B \otimes C) & & B \otimes (A \otimes C) \\ {\scriptstyle \beta_{A,B \otimes C}} \downarrow & & \downarrow {\scriptstyle 1_B \otimes \beta_{A,C}} \\ (B \otimes C) \otimes A & \xrightarrow[\alpha_{B,C,A}]{} & B \otimes (C \otimes A) \end{array}$$

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

2.2 Smash products

The model of the smash product we use here is given by the cofibre of the inclusion $A \vee B \hookrightarrow A \times B$, that is, the following (homotopy) pushout.

$$\begin{array}{ccc} A \vee B & \longrightarrow & A \times B \\ \downarrow & \ulcorner & \downarrow \\ 1 & \longrightarrow & A \wedge B \end{array}$$

For the sake of clarity, let us spell this out in detail by giving explicit names to all constructors of $A \wedge B$. We give the smash product the following self-contained definition.

Definition 5 (Smash products). *The smash product of two pointed types A and B is the HIT generated by:*

- *a point* $\star_\wedge : A \wedge B$,
- *points* $\langle a, b\rangle : A \wedge B$ *for every pair* $(a, b) : A \times B$,
- *paths* $\mathsf{push_l}(a) : \langle a, \star_B\rangle = \star_\wedge$ *for every point* $a : A$,
- *paths* $\mathsf{push_r}(b) : \langle \star_A, b\rangle = \star_\wedge$ *for every point* $b : B$,
- *a coherence* $\mathsf{push_{lr}} : \mathsf{push_l}(\star_A) = \mathsf{push_r}(\star_B)$.

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

Remark 6. *We remark that we could equivalently have defined the smash product by the following pushout.*

$$\begin{array}{ccc} A + B & \longrightarrow & A \times B \\ \downarrow & \ulcorner & \downarrow \\ 1 + 1 & \longrightarrow & A \wedge B \end{array}$$

This definition has the advantage of not having any 2-dimensional path constructors but has the disadvantage of having an additional point constructor. It turns out that Definition 5 suits our purposes better, so we stick with it. Alternatively, we could have defined the smash product to have $\langle \star_A, \star_B\rangle$ as its canonical basepoint. Such a definition is obtained by forming the HIT generated by:

- *points* $\langle a, b\rangle : A \wedge B$ *for every pair* $(a, b) : A \times B$,
- *paths* $\mathsf{push_l}(a) : \langle a, \star_B\rangle = \langle \star_A, \star_B\rangle$ *for every point* $a : A$,
- *paths* $\mathsf{push_r}(b) : \langle \star_A, b\rangle = \langle \star_A, \star_B\rangle$ *for every point* $b : B$,
- *a coherence* $\mathsf{push_{lr}} : \mathsf{push_l}(\star_A) = \mathsf{push_r}(\star_B)$,
- *a coherence* $\mathsf{push_{l\star}} : \mathsf{push_l}(\star_A) = \mathsf{refl}$.

It was suggested to us by an anonymous reviewer that this definition could allow certain statements in this paper to be expressed in a slightly less convoluted way (in particular Lemmas 19 and 20). We stick to Definition 5 as it is more commonly used in the HoTT literature but remark that the interested reader may find it enlightening to reinterpret the results of this paper in terms of the definition above.

Let us 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 $\star_f : f(\star_A) = \star_B$.

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

Definition 7. *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 on point constructors by:*

$$(f \wedge g)\,(\star_\wedge) = \star_\wedge$$
$$(f \wedge g)\,\langle a, b\rangle = \langle f(a), g(b)\rangle$$

and on (1-dimensional) path constructors by the following compositions:

$$\mathsf{ap}_{f\wedge g}(\mathsf{push}_\mathsf{l}(a)) \;=\; \langle f(a), g(\star_B)\rangle \xrightarrow{\mathsf{ap}_{\langle f(a),-\rangle}(\star_g)} \langle f(a), \star_D\rangle \xrightarrow{\mathsf{push}_\mathsf{l}(f(a))} \star_\wedge$$
$$\mathsf{ap}_{f\wedge g}(\mathsf{push}_\mathsf{r}(b)) \;=\; \langle f(\star_A), g(b)\rangle \xrightarrow{\mathsf{ap}_{\langle -,g(b)\rangle}(\star_f)} \langle \star_C, g(b)\rangle \xrightarrow{\mathsf{push}_\mathsf{r}(g(b))} \star_\wedge$$

The final case of the definition, that is, $\mathsf{ap}_{\mathsf{ap}_{f\wedge g}}(\mathsf{push}_{\mathsf{lr}})$, is a simple coherence which does not matter in the remainder of the paper. We take $f \wedge g$ 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.

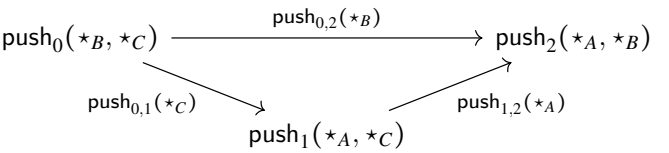
Proposition 8. *The swap map $A \times B \to B \times A$ induces a pointed equivalence $A \wedge B \simeq_\star B \wedge A$.*

3. Associativity

Proving that the smash product is associative is far less straightforward than proving that it is commutative. In fact, even the seemingly direct task of constructing the associator map is no mean feat. While associativity has already been verified by van Doorn (2018) and, using a computer-generated proof, by Brunerie (2018), let us give a direct construction of the equivalence. We do this because an explicit description makes the associator easier to trace when verifying, for example, the pentagon identity. For this purpose, let us introduce a new HIT capturing double smash products in a way which is neutral with respect to the distribution of parentheses.

Definition 9. *Given pointed types A, B and C, we define the type $\bigwedge(A, B, C)$ to be the HIT generated by:*

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



https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

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

$(A \wedge B) \wedge C$	$\to$	$\bigwedge(A, B, C)$
$\star_\wedge$	$\mapsto$	$\star_{2\wedge}$
$\langle \star_\wedge, c \rangle$	$\mapsto$	$\star_{2\wedge}$
$\langle \langle a, b \rangle, c \rangle$	$\mapsto$	$\langle a, b, c \rangle$
$\mathrm{ap}_{\langle -, c \rangle}(\mathrm{push}_1(a))$	$\mapsto$	$\mathrm{push}_1(a, c)$
$\mathrm{ap}_{\langle -, c \rangle}(\mathrm{push}_r(b))$	$\mapsto$	$\mathrm{push}_0(b, c)$
$\mathrm{ap}_{\mathrm{ap}_{\langle -, c \rangle}}(\mathrm{push}_{lr})$	$\mapsto$	$\mathrm{push}_{0,1}(c)$
$\mathrm{push}_l(\star_\wedge)$	$\mapsto$	refl
$\mathrm{push}_l\langle a, b \rangle$	$\mapsto$	$\mathrm{push}_2(a, b)$
$\mathrm{ap}_{\mathrm{push}_l}(\mathrm{push}_l(a))$	$\mapsto$	$push_{1,2}(a)$
$\mathrm{ap}_{\mathrm{push}_l}(\mathrm{push}_r(b))$	$\mapsto$	$push_{0,2}(b)$
$\mathrm{ap}_{\mathrm{ap}_{\mathrm{push}_l}}(\mathrm{push}_{lr})$	$\mapsto$	$\mathrm{push}_{0,1,2}$
$\mathrm{push}_r(c)$	$\mapsto$	refl
push_{lr}	$\mapsto$	refl

Let us verify that this actually captures a double smash product. What we need is an equivalence $(A \wedge B) \wedge C \simeq \bigwedge(A, B, C)$. The underlying map of this 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 only serves as an informal sketch of the function – 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; the interested reader is referred to the computer formalisation. We get the associativity of the smash product as a consequence.

Proposition 10. *There is a pointed equivalence* $\alpha_{A,B,C} : (A \wedge B) \wedge C \simeq_\star A \wedge (B \wedge C)$

Proof. Observe that $\bigwedge(A, B, C)$ is trivially invariant under permutation of the arguments in the sense that, for example, $\bigwedge(A, B, C) \simeq \bigwedge(B, C, A)$. This allows us to define $\alpha_{A,B,C}$ by the following composition of equivalences:

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

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

4. The Heuristic

Reasoning about functions defined over iterated smash products quickly gets out of hand. For instance, in order to verify 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 two such functions f and g, we have to construct, for instance, a dependent path:

$$\mathsf{ap}_{\mathsf{ap}_f \circ \mathsf{push}_l} \circ \mathsf{push}_l(\mathsf{push}_l(a)) \rightsquigarrow \mathsf{ap}_{\mathsf{ap}_g \circ \mathsf{push}_l} \circ \mathsf{push}_l(\mathsf{push}_l(a)) \tag{2}$$

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

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

problems are, in some sense, automatic. This was, in fact, suggested by Brunerie (2016) but was never proved nor 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 exact same HIT as $A \wedge B$ but with the $\mathsf{push}_{\mathsf{lr}}$ constructor removed. In other words, it is the following pushout.

$$\begin{array}{ccc} A + B & \longrightarrow & A \times B \\ \downarrow & \ulcorner & \downarrow \\ 1 & \longrightarrow & A \mathbin{\widetilde{\wedge}} B \end{array}$$

Let i be the obvious map $A \mathbin{\widetilde{\wedge}} B \to A \wedge B$.

Lemma 11. *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

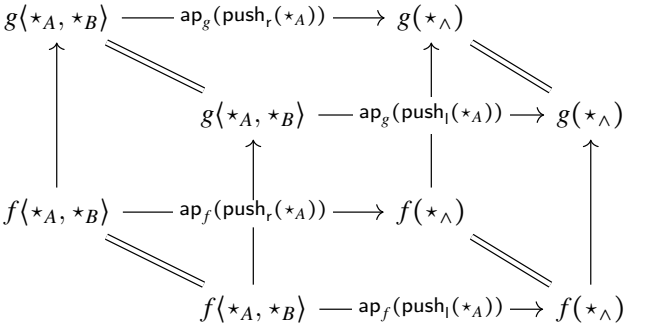
- a path $p : f(\star_\wedge) = g(\star_\wedge)$,
- a homotopy $h : ((a, b) : A \times B) \to f\langle a, b\rangle = g\langle a, b\rangle$,
- for each $a : A$, a filler $h_l(a)$ of the square

$$\begin{array}{ccc} g\langle a, \star_B\rangle & \xrightarrow{\mathsf{ap}_g(\mathsf{push}_{\mathsf{l}}(a))} & g(\star_\wedge) \\ {\scriptstyle h(a,\star_B)}\uparrow & & \uparrow{\scriptstyle p} \\ f\langle a, \star_B\rangle & \xrightarrow[\mathsf{ap}_f(\mathsf{push}_{\mathsf{l}}(a))]{} & f(\star_\wedge) \end{array}$$

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

$$\begin{array}{ccc} g\langle \star_A, b\rangle & \xrightarrow{\mathsf{ap}_g(\mathsf{push}_{\mathsf{r}}(b))} & g(\star_\wedge) \\ {\scriptstyle h(\star_A,b)}\uparrow & & \uparrow{\scriptstyle p} \\ f\langle \star_A, b\rangle & \xrightarrow[\mathsf{ap}_f(\mathsf{push}_{\mathsf{r}}(b))]{} & f(\star_\wedge) \end{array}$$

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



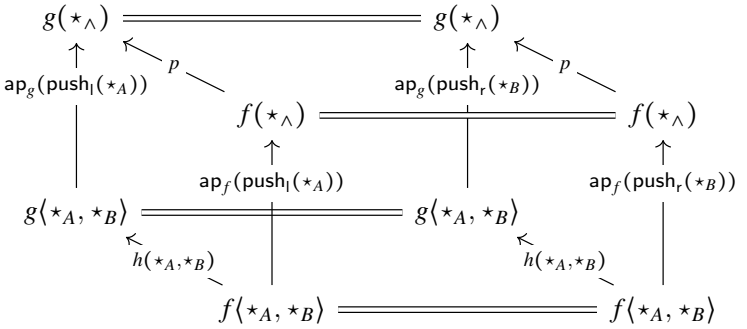
https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

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

$$\begin{array}{ccccc} g\langle \star_A, b\rangle & \xrightarrow{\mathsf{ap}_g(\mathsf{push}_{\mathsf{r}}(b))} & g(\star_\wedge) & = & g(\star_\wedge) \\ {\scriptstyle h(\star_A,b)}\uparrow & & \uparrow{\scriptstyle p} & & \uparrow{\scriptstyle p} \\ f\langle \star_A, b\rangle & \xrightarrow[\mathsf{ap}_f(\mathsf{push}_{\mathsf{r}}(b))]{} & f(\star_\wedge) & = & f(\star_\wedge) \end{array}$$

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



with $h_l(\star_A)$ and $h_r(\star_B)$ as left- and right-hand sides, the action of f and g on $\mathsf{push}_{\mathsf{lr}}$ on the front and back, 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. □

Remark 12. *Lemma 11 is a special case of the more general statement that i is a retraction (this was called to our attention by Dan Petersen). In fact, there is an equivalence* $A \mathbin{\widetilde{\wedge}} B \simeq S^1 \vee (A \wedge B)$ *under which the map i becomes the canonical retraction* $S^1 \vee (A \wedge B) \to A \wedge B$. *To see this, consider the following commutative diagram.*

$$\begin{array}{ccccc} 1 & \leftarrow & 2 & \xrightarrow{\mathsf{id}} & 2 \\ \uparrow & & \uparrow{\scriptstyle \mathsf{id}} & & \uparrow \\ 2 & \xleftarrow{\mathsf{id}} & 2 & \longrightarrow & A+B \\ \downarrow & & \downarrow & & \downarrow \\ 1 & \leftarrow & 1 & \longrightarrow & A\times B \end{array}$$

By taking pushouts of the rows, we produce the span $1 \leftarrow A + B \rightarrow A \times B$, *the pushout of which is* $A \mathbin{\widetilde{\wedge}} B$. *On the other hand, by taking pushouts of the columns, we produce the span* $S^1 \leftarrow 1 \rightarrow A \wedge B$, *the pushout of which is* $S^1 \vee (A \wedge B)$. *This yields the desired equivalence* $A \mathbin{\widetilde{\wedge}} B \simeq S^1 \vee (A \wedge B)$ *by the 3×3-lemma (introduced by* Brunerie 2016, *Lemma 1.8.3 with computer formalisation by* Pujet and Mörtberg 2020). *The fact that i factors in the appropriate manner follows by construction. Thus, we have given an alternative proof of Lemma 11.*

Lemma 11 is useful but does not get us all the way. Complicated paths like (2) still need to be constructed, regardless of what happens with the $\mathsf{push}_{\mathsf{lr}}$ constructor. To strengthen the principle,

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

we will need to introduce the concept of *homogeneous types* which, to the best of our knowledge, was first introduced (in HoTT) by Kraus (2013).

Definition 13. *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 by the following lemma which was first conjectured for Eilenberg–MacLane spaces in work leading up to Brunerie et al. (2022) and later proved and generalised by Cavallo (2021b) (and later further generalised by Buchholtz et al. 2023).

Lemma 14 (Cavallo). *Let $f, g : A \to_\star B$ and let B be 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 15. *Let $f, g : A \wedge B \to_\star C$ and let C be homogeneous. If $f\langle a, b\rangle = g\langle a, b\rangle$ for all $a : A$ and $b : B$, we obtain an equality of pointed functions $f = g$.*

The following form is useful when dealing with non-pointed functions.

Lemma 16. *Let C be an arbitrary type and suppose that we have two functions $f, g : A \wedge B \to C$ with $(C, f(\star_\wedge))$ homogeneous. If $f\langle a, b\rangle = g\langle a, b\rangle$ for all $a : A$ and $b : B$, then $f = g$.*

Proof. Let $h(a, b) : f\langle a, b\rangle = g\langle a, b\rangle$. We know that $g(\star_\wedge) = f(\star_\wedge)$ by the composite path

$$g(\star_\wedge) \xrightarrow{\mathsf{ap}_g(\mathsf{push}_\mathsf{l}(\star_A))^{-1}} g\langle \star_A, \star_B\rangle \xrightarrow{h(\star_A,\star_B)^{-1}} f\langle \star_A, \star_B\rangle \xrightarrow{\mathsf{ap}_f(\mathsf{push}_\mathsf{l}(\star_A))} f(\star_\wedge)$$

Hence, we may view both f and g as pointed functions $A \wedge B \to_\star (C, f(\star_\wedge))$. Now, Lemma 15 applies and, in particular, $f = g$ as plain functions. □

If we could apply Lemma 15 or 16 when proving the pentagon identity, we would be done immediately. Unfortunately, none of the types showing up in the statement of the pentagon identity are necessarily homogeneous. There is, however, some use for the lemmas. 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 $\mathsf{push}_\mathsf{l}/\mathsf{push}_\mathsf{r}$. The obvious definition of these maps (which we will spell out in Definition 17) will give us a version of Lemma 11 which tells us that if L_h and R_h are constant, then $f = g$ as plain functions. Now, if either A or B is another smash product, this *would* be a situation where Lemma 16 applies since $\Omega(C)$ (and indeed any path type) is homogeneous. This suggests that Lemma 16 may be used our advantage when dealing with iterated smash products. Let us try to spell this out more clearly. We can state the idea without any pointedness assumptions by simply replacing $\Omega(C)$ with the path type $f(\star_\wedge) = g(\star_\wedge)$ which is also homogeneous (and agrees with $\Omega(C)$ whenever f and g are pointed).

Definition 17. *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 functions $L_h : A \to f(\star_\wedge) = g(\star_\wedge)$ and $R_h : B \to f(\star_\wedge) = g(\star_\wedge)$ defined by:*

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

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

We may use L_h and R_h to give a compact induction principle for identities $f = g$. The following lemma is a direct rewriting of Lemma 11.

Lemma 18. *Let $f, g : A \wedge B \to C$. The following data yields an equality $f = g$.*

- *A homotopy $h : ((a, b) : A \times B) \to f\langle a, b\rangle = g\langle a, b\rangle$,*
- *paths $L_h = \mathsf{const}_{L_h(\star_A)}$ and $R_h = \mathsf{const}_{R_h(\star_B)}$.*

where const_y denotes the constant function, that is, $\mathsf{const}_y(x) = y$. If C is a pointed type, f and g are pointed functions and an equality $f = g$ of pointed functions is desired, a further coherence $\star_f = L_h(\star_A) \cdot \star_g$ is required.

The key idea now is, we stress again, to use Lemmas 18 and 15/16 iteratively to prove equalities of functions defined over iterated smash products. As a first example, let us consider the problem of proving an equality of functions $f = g$ where $f, g : (A \wedge B) \wedge C \to D$. The following lemma provides a good estimate of the minimal amount of data needed to construct such an equality.

Lemma 19. *For 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 $(a, c) : A \times C$, a filler of the square

$$\begin{array}{ccc}
f\langle \star_A, \star_B, c\rangle & \xrightarrow{h(\star_A, \star_B, c)} & g\langle \star_A, \star_B, c\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -, c\rangle}(\mathsf{push}_\mathsf{l}(\star_A))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -, c\rangle}(\mathsf{push}_\mathsf{l}(\star_A))^{-1}} \\
f\langle \star_\wedge, c\rangle & & g\langle \star_\wedge, c\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -, c\rangle}(\mathsf{push}_\mathsf{l}(a))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -, c\rangle}(\mathsf{push}_\mathsf{l}(a))} \\
f\langle a, \star_B, c\rangle & \xrightarrow[h(a, \star_B, c)]{} & g\langle a, \star_B, c\rangle
\end{array}$$

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

$$\begin{array}{ccc}
f\langle \star_A, \star_B, c\rangle & \xrightarrow{h(\star_A, \star_B, c)} & g\langle \star_A, \star_B, c\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -, c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -, c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}} \\
f\langle \star_\wedge, c\rangle & & g\langle \star_\wedge, c\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -, c\rangle}(\mathsf{push}_\mathsf{r}(b))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -, c\rangle}(\mathsf{push}_\mathsf{r}(b))} \\
f\langle \star_A, b, c\rangle & \xrightarrow[h(\star_A, b, c)]{} & g\langle \star_A, b, c\rangle
\end{array}$$

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

(iv) for every pair $(a, b) : A \times B$*, a filler of the square*

$$\begin{array}{ccc}
f\langle \star_A, \star_B, \star_C\rangle & \xrightarrow{h(\star_A,\star_B,\star_C)} & g\langle \star_A, \star_B, \star_C\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B)^{-1}))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B)^{-1}))} \\
f\langle \star_\wedge, \star_C\rangle & & g\langle \star_\wedge, \star_C\rangle \\
{\scriptstyle \mathsf{ap}_f(\mathsf{push}_\mathsf{l}(\star_\wedge)^{-1})}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_\mathsf{l}(\star_\wedge)^{-1})} \\
f(\star_\wedge) & & g(\star_\wedge) \\
{\scriptstyle \mathsf{ap}_f(\mathsf{push}_\mathsf{l}\langle a,b\rangle)}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_\mathsf{l}\langle a,b\rangle)} \\
f\langle a, b, \star_C\rangle & \xrightarrow[h(a,b,\star_C)]{} & g\langle a, b, \star_C\rangle
\end{array}$$

(v) for every point $c : C$*, a filler of the square*

$$\begin{array}{ccc}
f\langle \star_A, \star_B, \star_C\rangle & \xrightarrow{h(\star_A,\star_B,\star_C)} & g\langle \star_A, \star_B, \star_C\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C\rangle}(\mathsf{push}_\mathsf{r}(\star_B))^{-1}} \\
f\langle \star_\wedge, \star_C\rangle & & g\langle \star_\wedge, \star_C\rangle \\
{\scriptstyle \mathsf{ap}_f(\mathsf{push}_\mathsf{l}(\star_\wedge))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_\mathsf{l}(\star_\wedge))^{-1}} \\
f(\star_\wedge) & & g(\star_\wedge) \\
{\scriptstyle \mathsf{ap}_f(\mathsf{push}_\mathsf{r}(c))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_\mathsf{r}(c))} \\
f\langle \star_\wedge, c\rangle & & g\langle \star_\wedge, c\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,c\rangle}(\mathsf{push}_\mathsf{r}(\star_B))} \\
f\langle \star_A, \star_B, c\rangle & \xrightarrow[h(\star_A,\star_B,c)]{} & g\langle \star_A, \star_B, c\rangle
\end{array}$$

Proof sketch. Suppose we have the given data. We apply Lemma 18 to f and g. This breaks the proof up into two 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 fix $c : C$ and note that we may now apply Lemma 18 again: this time to the functions $f(-, c)$ and $g(-, c)$. This gives us two 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 finally need to show that $L_{h(-,c)}$ and $R_{h(-,c)}$ are constant. This boils down to providing fillers of the squares which we assumed in (ii) and (iii).
- We then need to show that L_k and R_k are constant. To show that L_k is constant, we apply Lemma 16. This is justified since the codomain of L_k 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 by assumption (iv). Note that this is where the explosion of complexity would normally happen but, thanks to Lemma 16, we completely avoid having to verify any higher coherences. For R_k, we again unfold the definitions to see that the path required is provided by (v). □

For completeness, let us state the corresponding lemma for functions $f, g : ((A \wedge B) \wedge C) \wedge D \to E$, as these appear in the pentagon identity. The proof is by Lemmas 18,16 and 19, following

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

the exact same line of attack as before. We acknowledge that the following result is highly technical but stress that the interesting aspect of it is not the exact details of the statement; rather, we 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 naive inductive proof.

Lemma 20. *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 $(a, b, c) : A \times B \times C$, a filler of the square

$$\begin{array}{ccc}
f\langle \star_A, \star_B, \star_C \star_D\rangle & \xrightarrow{h(\star_A,\star_B,\star_C\star_D)} & g\langle \star_A, \star_B, \star_C \star_D\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A)))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A)))^{-1}} \\
f\langle \star_\wedge, \star_C, \star_D\rangle & & g\langle \star_\wedge, \star_C, \star_D\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_D\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}} \\
f\langle \star_\wedge, \star_D\rangle & & g\langle \star_\wedge, \star_D\rangle \\
{\scriptstyle \mathsf{ap}_{f}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)))^{-1}} \\
f(\star_\wedge) & & g(\star_\wedge) \\
{\scriptstyle \mathsf{ap}_{f}(\mathsf{push}_{\mathsf{l}}\langle a,b,c\rangle)}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g}(\mathsf{push}_{\mathsf{l}}\langle a,b,c\rangle)} \\
f\langle a, b, c, \star_D\rangle & \xrightarrow[h(a,b,c,\star_D)]{} & g\langle a, b, c, \star_D\rangle
\end{array}$$

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

$$\begin{array}{ccc}
f\langle \star_A, \star_B, \star_C, d\rangle & \xrightarrow{h(\star_A,\star_B,\star_C)} & g\langle \star_A, \star_B, \star_C, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B)^{-1}))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C,d\rangle}(\mathsf{push}_{\mathsf{r}}(\star_B)^{-1}))} \\
f\langle \star_\wedge, \star_C, d\rangle & & g\langle \star_\wedge, \star_C, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)^{-1})}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_\wedge)^{-1})} \\
f\langle \star_\wedge, d\rangle & & g\langle \star_\wedge, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}\langle a,b\rangle)}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push}_{\mathsf{l}}\langle a,b\rangle)} \\
f\langle a, b, \star_C, d\rangle & \xrightarrow[h(a,b,\star_C)]{} & g\langle a, b, \star_C, d\rangle
\end{array}$$

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

$$\begin{array}{ccc}
f\langle \star_A, \star_B, c, d\rangle & \xrightarrow{h(\star_A,\star_B,c,d)} & g\langle \star_A, \star_B, c, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(\star_A))^{-1}} \\
f\langle \star_\wedge, c, d\rangle & & g\langle \star_\wedge, c, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(a))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push}_{\mathsf{l}}(a))} \\
f\langle a, \star_B, c, d\rangle & \xrightarrow[h(a,\star_B,c,d)]{} & g\langle a, \star_B, c, d\rangle
\end{array}$$

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

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

$$\begin{array}{ccc}
f\langle \star_A, \star_B, c, d\rangle & \xrightarrow{h(\star_A,\star_B,c,d)} & g\langle \star_A, \star_B, c, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push_r}(\star_B))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push_r}(\star_B))^{-1}} \\
f\langle \star_\wedge, c, d\rangle & & g\langle \star_\wedge, c, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push_r}(b))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push_r}(b))} \\
f\langle \star_A, b, c, d\rangle & \xrightarrow[h(\star_A,b,c,d)]{} & g\langle \star_A, b, c, d\rangle
\end{array}$$

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

$$\begin{array}{ccc}
f\langle \star_A, \star_B, \star_C, d\rangle & \xrightarrow{h(\star_A,\star_B,\star_C,d)} & g\langle \star_A, \star_B, \star_C\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C,d\rangle}(\mathsf{push_r}(\star_B))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C,d\rangle}(\mathsf{push_r}(\star_B))^{-1}} \\
f\langle \star_\wedge, \star_C, d\rangle & & g\langle \star_\wedge, \star_C\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push_l}(\star_\wedge))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push_l}(\star_\wedge))^{-1}} \\
f\langle \star_\wedge, d\rangle & & g\langle \star_\wedge, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push_r}(c))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push_r}(c))} \\
f\langle \star_\wedge, c, d\rangle & & g\langle \star_\wedge, c, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,c,d\rangle}(\mathsf{push_r}(\star_B))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,c,d\rangle}(\mathsf{push_r}(\star_B))} \\
f\langle \star_A, \star_B, c, d\rangle & \xrightarrow[h(\star_A,\star_B,c,d)]{} & g\langle \star_A, \star_B, c, d\rangle
\end{array}$$

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

$$\begin{array}{ccc}
f\langle \star_A, \star_B, \star_C, \star_D\rangle & \xrightarrow{h(\star_A,\star_B,\star_C,\star_D)} & g\langle \star_A, \star_B, \star_C, \star_D\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C,\star_D\rangle}(\mathsf{push_l}(\star_A)))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C,\star_D\rangle}(\mathsf{push_l}(\star_A)))^{-1}} \\
f\langle \star_\wedge, \star_C, \star_D\rangle & & g\langle \star_\wedge, \star_C, \star_D\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_D\rangle}(\mathsf{push_l}(\star_\wedge)))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_D\rangle}(\mathsf{push_l}(\star_\wedge)))^{-1}} \\
f\langle \star_\wedge, \star_D\rangle & & g\langle \star_\wedge, \star_D\rangle \\
{\scriptstyle \mathsf{ap}_{f}(\mathsf{push_l}(\star_\wedge)))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g}(\mathsf{push_l}(\star_\wedge)))^{-1}} \\
f(\star_\wedge) & & g(\star_\wedge) \\
{\scriptstyle \mathsf{ap}_{f}(\mathsf{push_r}(d))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g}(\mathsf{push_r}(d))} \\
f\langle \star_\wedge, d\rangle & & g\langle \star_\wedge, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,d\rangle}(\mathsf{push_l}(\star_\wedge)))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,d\rangle}(\mathsf{push_l}(\star_\wedge)))} \\
f\langle \star_\wedge, \star_C, d\rangle & & g\langle \star_\wedge, \star_C, d\rangle \\
{\scriptstyle \mathsf{ap}_{f\langle -,\star_C,d\rangle}(\mathsf{push_l}(\star_A)))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{g\langle -,\star_C,d\rangle}(\mathsf{push_l}(\star_A)))} \\
f\langle \star_A, \star_B, \star_C, d\rangle & \xrightarrow[h(\star_A,\star_B,\star_C,d)]{} & g\langle \star_A, \star_B, \star_C, d\rangle
\end{array}$$

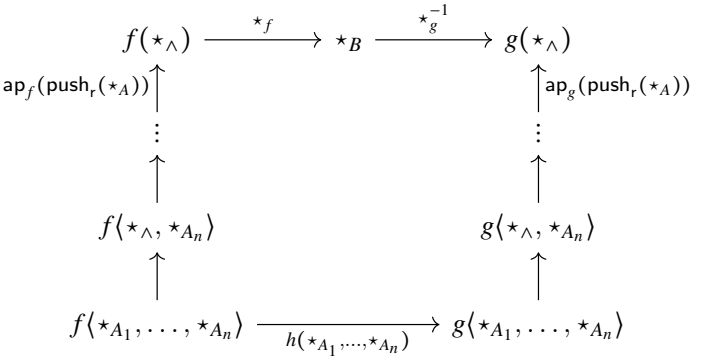
https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

Let us make three observations about Lemmas 19 and 20:

1. In both statements, the different pieces of data are almost completely mutually independent. The only meaningful choice we can make is that of the 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 concrete 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 defined entirely in terms of applications of f and g on push_l and push_r – something we can usually simply unfold to something (hopefully) simple.
3. Going from Lemmas 19–20, we see that only two additional assumptions are needed. If we were 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 naive proof by a deep smash product induction, the amount of data needed would grow exponentially.

While it is possible to generalise Lemmas 19 and 20 to a statement concerning n-fold smash products, let us, for now, only summarise the fundamental idea behind Lemmas 19 and 20 in terms of an informal heuristic.

Heuristic 21. *To show that two functions $f, g : \bigwedge_{i \leq n} A_i \to B$ are equal, it suffices, by iterative application Lemmas 18, 16 and 14, 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 it is coherent with f and g on any single application of push_l or push_r (e.g., $\mathsf{ap}_{\langle -, x_{i+1}, \dots, x_n\rangle}(\mathsf{push}_\mathsf{l}\langle x_1, \dots, x_{i-1}\rangle)$). Furthermore, if an equality of pointed functions is required, we need to provide a filler of the following square:*



The reader who is looking for a precise mathematical statement of the above is referred to Section 6 where we also introduce some additional machinery required to make the idea formal. For now, we will settle for this informal heuristic – in practice, when working with functions over a small fixed number of smash products, we do not quite need the full strength of a general theorem.

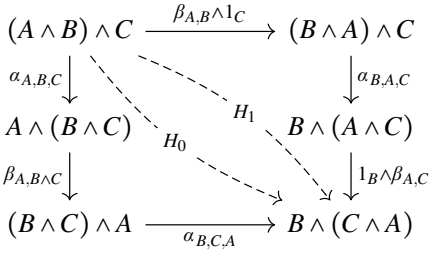
5. The Symmetric Monoidal Structure

Let us reap the fruits of our labour and show that the smash product defines a (1-coherent) symmetric monoidal product on the universe of pointed types. We will not verify all axioms

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

here, since this is neither very instructive nor very interesting. Instead, we sketch the proofs of the two most technical properties and refer the interested reader to the computer formalisation.

Proposition 22. *The smash product satisfies the hexagon identity, that is, 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 following hexagon.*



Proof sketch. We show the statement by an application of our heuristic which, in this case, takes the form of Lemma 19. We provide the data as follows:

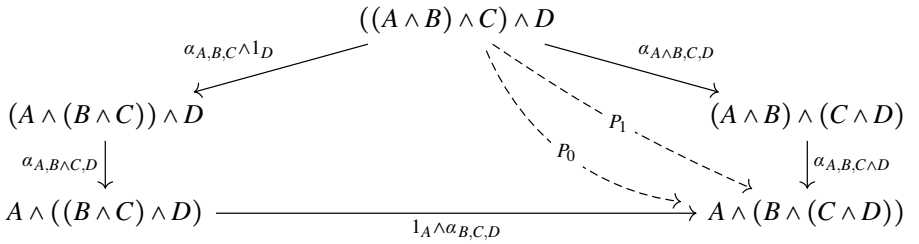
1. For the homotopy $h(a, b, c) : H_0\langle a, b, c\rangle = H_1\langle a, b, c\rangle$, we simply choose $h(a, b, c) := \mathsf{refl}$, since both sides compute to $\langle b, c, a\rangle$.
2. For example, the fourth datum in Lemma 19 computes to[1] the following square-filling problem:

$$\begin{array}{ccc}
\langle \star_B, \star_C, \star_A\rangle & \xrightarrow{\mathsf{refl}} & \langle \star_B, \star_C, \star_A\rangle \\
{\scriptstyle \mathsf{push_l}(\star_B)^{-1}\cdot \mathsf{ap}_{\langle \star_B, -\rangle}(\mathsf{push_l}(\star_C))^{-1}}\uparrow & & \uparrow{\scriptstyle \mathsf{push_l}(\star_B)^{-1}\cdot \mathsf{ap}_{\langle \star_B, -\rangle}(\mathsf{push_l}(\star_C))^{-1}} \\
\wedge C & & \wedge C \\
{\scriptstyle \mathsf{refl}}\uparrow & & \uparrow{\scriptstyle \mathsf{refl}} \\
\wedge C & & \wedge C \\
{\scriptstyle \mathsf{ap}_{\langle b, -\rangle}(\mathsf{push_r}(a))\cdot \mathsf{push_l}(b)}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_{\langle b, -\rangle}(\mathsf{push_r}(a))\cdot \mathsf{push_l}(b)} \\
\langle b, \star_C, a\rangle & \xrightarrow[\mathsf{refl}]{} & \langle b, \star_C, a\rangle
\end{array}$$

   which 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 Heuristic 21. 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 23. *The pentagon identity holds for the smash product, that is, 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 following pentagon.*

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press



Proof. The statement follows easily by the heuristic, which in this case corresponds to Lemma 20. The proof is identical to the proof of Proposition 22 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 20 become trivial. □

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

Theorem 24. *The universe of pointed types forms a symmetric monoidal wild category with the smash product as tensor product.*

5.1 Back to Brunerie's thesis

A first consequence of the fact that Theorem 24 finally has a complete and computer formalised proof is the correctness of Brunerie's PhD thesis (2016). While a computer formalisation of the main results of the thesis was presented by Ljungström and Mörtberg (2023), this formalisation did not stay completely true to Brunerie's original proof. Indeed, certain proofs and constructions were reworked in order not to rely on results concerning smash products. Most notably, the *cup product* was redefined using an alternative definition by Brunerie et al. (2022, Section 4.1). In Brunerie's thesis, the cup product (on integral Eilenberg–MacLane spaces) is defined as a map $\smile : K(\mathbb{Z}, n) \wedge K(\mathbb{Z}, m) \to K(\mathbb{Z}, n+m)$ where, for $n, m \geq 1$, we have $K(\mathbb{Z}, n) := \|S^n\|_n$, that is, the n-truncation of the n-sphere. Here, we define $S^n := \Sigma^{n+1}(\emptyset)$, that is, it is the $(n+1)$-fold suspension of the empty type, where, recall, the suspension of a type A, is the HIT $\Sigma(A)$ generated by

- two points $\mathsf{north}, \mathsf{south} : \Sigma(A)$,
- paths $\mathsf{merid}\,(a) : \mathsf{north} = \mathsf{south}$ for each $a : A$.

Brunerie's construction of the cup product is by means of a lift from the corresponding map on spheres.

$$\begin{array}{ccc} S^n \wedge S^m & \xrightarrow{\wedge_{n,m}} & S^{n+m} \\ {\scriptstyle |-|\wedge|-|}\downarrow & & \downarrow{\scriptstyle |-|} \\ K(\mathbb{Z}, n) \wedge K(\mathbb{Z}, m) & \dashrightarrow_{\smile} & K(\mathbb{Z}, n+m) \end{array}$$

Above, the map $\wedge_{n,m} : S^n \wedge S^m \to S^{n+m}$ is the canonical equivalence defined in, for example, Brunerie (2016, Proposition 4.2.2). With this construction, most properties of the cup product can be shown by showing that the corresponding properties hold for $\wedge_{n,m}$. In particular, graded-commutativity and associativity follow, respectively, from the following two propositions.

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

Proposition 25 (Brunerie 2016, Proposition 4.2.4). *The following diagram commutes.*

$$\begin{array}{ccc} S^n \wedge S^m & \longrightarrow & S^m \wedge S^n \\ {\scriptstyle \wedge_{n,m}}\downarrow & & \downarrow{\scriptstyle \wedge_{m,n}} \\ S^{n+m} & \xrightarrow[(-1)^{nm}]{} & S^{n+m} \end{array}$$

where $(-1) : S^n \to S^n$ *is defined by sending* merid (a) *to* merid $(a)^{-1}$.

Proposition 26 (Brunerie 2016, Proposition 4.2.2). *The following diagram commutes.*

$$\begin{array}{ccccc} (S^n \wedge S^m) \wedge S^k & \xrightarrow{(\wedge_{n,m})\wedge 1_{S^k}} & S^{n+m} \wedge S^k & & \\ & & & \searrow^{\wedge_{n+m,k}} & \\ {\scriptstyle \alpha_{S^n,S^m,S^k}}\downarrow & & & & S^{n+n+k} \\ & & & \nearrow_{\wedge_{n,m+k}} & \\ S^n \wedge (S^m \wedge S^k) & \xrightarrow[1_{S^n}\wedge(\wedge_{m,k})]{} & S^n \wedge S^{m+k} & & \end{array}$$

These results are proved by Brunerie using proofs which rely on the symmetric monoidal structure of the smash product. While Proposition 25 should be provable using the partial proof of symmetric monoidality due to van Doorn (2018, Section 4.3), Brunerie's proof of Proposition 26 relies on the pentagon identity which, until now, has been open. Thus, with Theorem 24 we can finally conclude the following.

Theorem 27. *The cup product, as constructed in* Brunerie (2016, *Definition 5.1.6), forms a graded-commutative and associative multiplication* $K(\mathbb{Z}, n) \wedge K(\mathbb{Z}, m) \to K(\mathbb{Z}, n + m)$.

We remark that we have not provided a computer formalisation of the above since there already exist well-developed computer formalisations of the cup product and cohomology rings (Brunerie et al. 2022; Lamiaux et al. 2023; Ljungström and Mörtberg 2024).

6. A Formal Statement of the Heuristic

In a classical setting, the iterated smash product $\bigwedge_{i\leq n} A_i$ can be described as the quotient space $(A_0 \times \cdots \times A_n)/\mathsf{FW}_{i\leq n}(A_i)$ where

$$\mathsf{FW}_{i\leq n}(A_i) := \{(a_0, \ldots, a_n) \mid a_i = \star_{A_i} \text{ for some } i\} \subset A_0 \times \cdots \times A_n$$

It is unclear how to mimic this construction in HoTT in a useful way, as the naive definition forces us to add, in a systematic way, an exponential number of higher coherences. However, the naive definition may still prove useful. Here, by 'naive definition', we mean the disjoint union

$$\mathsf{FS}_{i\leq n}(A_i) := \bigsqcup_{i\leq n} \left(A_0^{(i)} \times \cdots \times A_n^{(i)}\right) \qquad \text{where } A_j^{(i)} = \begin{cases} A_j & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

While the cofibre of the canonical map $\mathsf{FS}_{i\leq n}(A_i) \to A_0 \times \cdots \times A_n$ does not quite give us $\bigwedge_{i\leq n} A_i$, it *does*, however, provide a useful approximation. This is, in fact, precisely the content of our heuristic. We will soon make this precise. Before this, let us give a less set-theoretic definition of $\mathsf{FS}_{i\leq n}(A_i)$ which is easier to work with in HoTT.

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

Definition 28. *Given pointed types $A_0, \dots, A_n$, we define $\mathsf{FS}_{i\leq n}(A_i)$ by induction on n as follows.*

$$\mathsf{FS}_{i\leq n}(A_i) := \begin{cases} 1 & \text{if } n = 0 \\ (\mathsf{FS}_{i\leq n-1}(A_i) \times A_n) + (A_0 \times \cdots \times A_{n-1}) & \text{if } n > 0 \end{cases}$$

Let us, for clarity, explicitly describe the canonical map $\gamma_n : \mathsf{FS}_{i\leq n}(A_i) \to A_0 \times \cdots \times A_n$. It is recursively defined with $\gamma_0 : 1 \to A_0$ picking out the basepoint $\star_{A_0}$ and

$$\gamma_n : (\mathsf{FS}_{i\leq n-1}(A_i) \times A_n) + (A_0 \times \cdots \times A_{n-1}) \to A_0 \times \cdots \times A_n$$

being defined by $\gamma_n = \gamma_n^{(1)} + \gamma_n^{(2)}$ where

$$\gamma_n^{(1)}(x, a_n) = (\gamma_{n-1}(x), a_n)$$
$$\gamma_n^{(2)}(a_0, \dots, a_{n-1}) = (a_0, \dots, a_{n-1}, \star A_n)$$

Definition 29. *Given pointed types $A_0, \dots, A_n$, we define $\widetilde{\bigwedge}_{i\leq n} A_i$ to be the cofibre of $\gamma_n : \mathsf{FS}_{i\leq n}(A_i) \to A_0 \times \cdots \times A_n$, that is, the following pushout.*

$$\begin{array}{ccc} \mathsf{FS}_{i\leq n}(A_i) & \xrightarrow{\gamma_n} & A_0 \times \cdots \times A_n \\ \downarrow & \ulcorner & \downarrow \\ 1 & \longrightarrow & \widetilde{\bigwedge}_{i\leq n} A_i \end{array}$$

We give its constructors explicit names and write

- $\star_{\widetilde{\wedge}}$ *for its basepoint,*
- $\langle a_0, \dots, a_n \rangle$ *for its underlying points,*
- $\mathsf{push}^{(1)}(x) : \langle \gamma_n^{(1)}(x) \rangle = \star_{\widetilde{\wedge}}$ *for the first coherence given by the pushout square,*
- $\mathsf{push}^{(2)}(x) : \langle \gamma_n^{(2)}(x) \rangle = \star_{\widetilde{\wedge}}$ *for the second coherence given by the pushout square.*

It is easy to see that the composition $\mathsf{FS}_{i\leq n}(A_i) \to A_0 \times \cdots \times A_n \to \bigwedge_{i\leq n} A_i$ is null-homotopic. Hence, there is a basepoint preserving inclusion (of constructors) $\iota : \widetilde{\bigwedge}_{i\leq n} A_i \to \bigwedge_{i\leq n} A_i$ s.t. $\iota\langle a_0, \dots, a_n \rangle = \langle a_0, \dots, a_n \rangle$.

We also remark that there is an inclusion $\uparrow : \left(\widetilde{\bigwedge}_{i\leq n-1} A_i\right) \times A_n \to \widetilde{\bigwedge}_{i\leq n} A_i$. It is defined by:

$$\uparrow (\star_{\widetilde{\wedge}}, a_n) := \star_{\widetilde{\wedge}}$$
$$\uparrow (\langle a_0, \dots, a_{n-1} \rangle, a_n) := \langle a_0, \dots, a_n \rangle$$
$$\mathsf{ap}_{\uparrow(-,a_n)}(\mathsf{push}(x)) := \mathsf{push}^{(1)}(x, a_n)$$

Furthermore, $\uparrow$ commutes with ι in the sense that, for $x : \widetilde{\bigwedge}_{i\leq n-1} A_i$ and $a_n : A_n$, we have the following path (in $\bigwedge_{i\leq n} A_i$):

$$\uparrow^{\mathsf{coh}} (x, a_n) \ : \iota (\uparrow (x, a_n)) = \langle \iota(x), a_n \rangle$$

One constructs $\uparrow^{\mathsf{coh}} (x, a_n)$ very directly by induction on x. Let us give the construction when x is a point constructor. When $x := \langle a_0, \dots, a_{n-1} \rangle$, the statement holds by refl. When $x := \star_{\widetilde{\wedge}}$, the

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

type of $\uparrow^{\mathsf{coh}}(x, a_n)$ reduces to $\star_\wedge = \langle \star_\wedge, a_n \rangle$ which holds by $\mathsf{push}_{\mathsf{r}}(a_n)^{-1}$. Hence, we define

$$\uparrow^{\mathsf{coh}}(\langle a_0, \ldots, a_{n-1} \rangle, a_n) := \mathsf{refl} \tag{3}$$

$$\uparrow^{\mathsf{coh}}(\star_{\widetilde{\wedge}}, a_n) := \mathsf{push}_{\mathsf{r}}(a_n)^{-1} \tag{4}$$

The higher coherence is very straightforward and we omit it for the sake of readability. We are now ready for the key result of the section. It makes precise to what extent $\widetilde{\bigwedge}_{i \leq n} A_i$ approximates $\bigwedge_{i \leq n} A_i$ and may be seen as a formal version of our heuristic. We will later restate the result in a less technical and more self-contained form.

Theorem 30. *Let $f, g : \wedge_{i \leq n} A_i \to B$. Given a homotopy $\widetilde{p}_n : (x : \widetilde{\bigwedge}_{i \leq n} A_i) \to f(\iota(x)) = g(\iota(x))$, there is a homotopy $p_n : (x : \wedge_{i \leq n} A_i) \to f(x) = g(x)$. Furthermore, p_n satisfies:*

$$p_n(\star_\wedge) = \widetilde{p}_n(\star_{\widetilde{\wedge}}) \tag{5}$$

$$p_n\langle a_0, \ldots, a_n \rangle = \widetilde{p}_n\langle a_0, \ldots, a_n \rangle \tag{6}$$

Proof. We proceed by induction on n. When $n = 0$, the map ι is simply the canonical equivalence between the cofibre of the unique pointed map $1 \to_\star A_0$ and A_0 itself, and the theorem is trivial.

Let $n \geq 1$ and let f, g and $\widetilde{p}_n$ be as in the theorem statement. Let us, for clarity, rewrite the domain of f and g as the binary smash product $\left(\bigwedge_{i \leq n-1} A_i\right) \wedge A_n$. In order to construct $p_n : (x : \left(\bigwedge_{i \leq n-1} A_i\right) \wedge A_n) \to f(x) = g(x)$, it suffices, by Lemma 11, to define

(a) a path $p_n(\star_\wedge) : f(\star_\wedge) = g(\star_\wedge)$,
(b) for $a_n : A_n$, homotopies $p_n\langle -, a_n \rangle : (x : \bigwedge_{i \leq n-1} A_i) \to f\langle x, y \rangle = g\langle x, y \rangle$,
(c) For each $a_n : A_n$, a filler of the square

$$\begin{array}{ccc} f(\star_\wedge) & \xrightarrow{p_n(\star_\wedge)} & g(\star_\wedge) \\ {\scriptstyle \mathsf{ap}_f(\mathsf{push}_{\mathsf{r}}(a_n))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_{\mathsf{r}}(a_n))} \\ f\langle \star_\wedge, a_n \rangle & \xrightarrow[p_n\langle \star_\wedge, a_n \rangle]{} & g\langle \star_\wedge, a_n \rangle \end{array}$$

(d) For each $x : \bigwedge_{i \leq n-1} A_i$, a filler of the square

$$\begin{array}{ccc} f(\star_\wedge) & \xrightarrow{p_n(\star_\wedge)} & g(\star_\wedge) \\ {\scriptstyle \mathsf{ap}_f(\mathsf{push}_{\mathsf{l}}(x))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_{\mathsf{l}}(x))} \\ f\langle x, \star_{A_n} \rangle & \xrightarrow[p_n\langle x, \star_{A_n} \rangle]{} & g\langle x, \star_{A_n} \rangle \end{array}$$

For (a), we know that $\iota(\star_{\widetilde{\wedge}}) := \star_\wedge$ and thus $\widetilde{p}_n(\star_{\widetilde{\wedge}}) : f(\star_\wedge) = g(\star_\wedge)$. Consequently, we may simply set $p_n(\star_\wedge) := \widetilde{p}_n(\star_{\widetilde{\wedge}})$. Doing so, (5) holds definitionally. For (b), we fix $a_n : A_n$. By the induction hypothesis, it suffices to construct $p_n\langle \iota(x), a_n \rangle : f\langle \iota(x), y \rangle = g\langle \iota(x), y \rangle$ for $x : \widetilde{\bigwedge}_{i \leq n-1} A_i$. We do this by the composite path

$$f\langle \iota(x), a_n \rangle \xrightarrow{\mathsf{ap}_f(\uparrow^{\mathsf{coh}}(x, a_n))^{-1}} f(\iota(\uparrow(x, a_n))) \xrightarrow{\widetilde{p}_n(\uparrow(x, a_n))} g(\iota(\uparrow(x, a_n))) \xrightarrow{\mathsf{ap}_g(\uparrow^{\mathsf{coh}}(x, a_n))} g\langle \iota(x), a_n \rangle$$

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

This completes the construction of $p_n\langle -, a_n\rangle$. Note that, since $p_n\langle -, a_n\rangle$ was constructed using the induction hypothesis, we also get the following from (5):

$$\begin{aligned} p_n\langle \star_\wedge, a_n\rangle &= \mathsf{ap}_f(\uparrow^{\mathsf{coh}}(\star_\wedge, a_n))^{-1} \cdot \widetilde{p}_n(\uparrow(\star_\wedge, a_n)) \cdot \mathsf{ap}_g(\uparrow^{\mathsf{coh}}(\star_\wedge, a_n)) && \text{by (5)} \\ &= \mathsf{ap}_f(\mathsf{push}_\mathsf{r}(a_n)) \cdot p_n(\star_\wedge) \cdot \mathsf{ap}_g(\mathsf{push}_\mathsf{r}(a_n))^{-1} && \text{by (4)} \end{aligned}$$

and the following from (6):

$$\begin{aligned} p_n\langle \vec{a}_{n-1}, a_n\rangle &= \mathsf{ap}_f(\uparrow^{\mathsf{coh}}(\vec{a}_{n-1}, a_n))^{-1} \cdot \widetilde{p}_n(\uparrow(\vec{a}_{n-1}, a_n)) \cdot \mathsf{ap}_g(\uparrow^{\mathsf{coh}}(\vec{a}_{n-1}, a_n)) && \text{by (6)} \\ &= \mathsf{refl} \cdot \widetilde{p}_n(\vec{a}_{n-1}, a_n) \cdot \mathsf{refl} && \text{by (3)} \\ &= \widetilde{p}_n(\vec{a}_{n-1}, a_n) \end{aligned}$$

where $\vec{a}_{n-1}$ is short for $\langle a_0, \dots, a_{n-1}\rangle$. So, to summarise, we have the following two identities for $p\langle -, a_n\rangle$:

$$p_n\langle \star_\wedge, a_n\rangle = \mathsf{ap}_f(\mathsf{push}_\mathsf{r}(a_n)) \cdot p_n(\star_\wedge) \cdot \mathsf{ap}_g(\mathsf{push}_\mathsf{r}(a_n))^{-1} \tag{7}$$

$$p_n\langle\langle a_0, \dots, a_{n-1}\rangle, a_n\rangle = \widetilde{p}_n\langle a_0, \dots, a_{n-1}, a_n\rangle \tag{8}$$

Note that, in particular, the latter identity verifies (6). Let us now turn to (c). By (7), we simply need to fill the square

$$\begin{array}{ccc} f(\star_\wedge) & \xrightarrow{\quad p_n(\star_\wedge) \quad} & g(\star_\wedge) \\ {\scriptstyle \mathsf{ap}_f(\mathsf{push}_\mathsf{r}(a_n))}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_\mathsf{r}(a_n))} \\ f\langle \star_\wedge, a_n\rangle & \xrightarrow[\mathsf{ap}_f(\mathsf{push}_\mathsf{r}(a_n)) \cdot p_n(\star_\wedge) \cdot \mathsf{ap}_g(\mathsf{push}_\mathsf{r}(a_n))^{-1}]{} & g\langle \star_\wedge, a_n\rangle \end{array}$$

which is entirely trivial by definition of path composition.

Finally, let us provide the data asked for in (d). Filling the square is equivalent to constructing, a homotopy $(x : \bigwedge_{i\le n-1} A_i) \to \mathsf{left}(x) = \mathsf{right}(x)$ where $\mathsf{left}, \mathsf{right} : \bigwedge_{i\le n-1} A_i \to f(\star_\wedge) = g(\star_\wedge)$ are defined by:

$$\begin{aligned} \mathsf{left}(x) &:= p_n(\star_\wedge) \\ \mathsf{right}(x) &:= \mathsf{ap}_f(\mathsf{push}_\mathsf{l}(x))^{-1} \cdot p_n\langle x, \star_{A_n}\rangle \cdot \mathsf{ap}_g(\mathsf{push}_\mathsf{l}(x)) \end{aligned}$$

The codomain $f(\star_\wedge) = g(\star_\wedge)$ is homogeneous (with $\mathsf{left}(\star_\wedge)$ as basepoint) and thus Lemma 16 applies. Hence, we only need to show that

$$\mathsf{left}\langle a_0, \dots, a_{n-1}\rangle = \mathsf{right}\langle a_0, \dots, a_{n-1}\rangle$$

Let us rewrite $p_n\langle -, \star_{A_n}\rangle$ according to (8) and transform the above back into square form. The problem is to fill the square

$$\begin{array}{ccc} f(\star_\wedge) & \xrightarrow{\quad \widetilde{p}_n(\star_\wedge) \quad} & g(\star_\wedge) \\ {\scriptstyle \mathsf{ap}_f(\mathsf{push}_\mathsf{l}\langle a_0,\dots,a_{n-1}\rangle)}\uparrow & & \uparrow{\scriptstyle \mathsf{ap}_g(\mathsf{push}_\mathsf{l}\langle a_0,\dots,a_{n-1}\rangle)} \\ f\langle\langle a_0, \dots, a_{n-1}\rangle, \star_{A_n}\rangle & \xrightarrow[\widetilde{p}_n\langle a_0,\dots,a_{n-1},\star_{A_n}\rangle]{} & g\langle\langle a_0, \dots, a_{n-1}\rangle, \star_{A_n}\rangle \end{array}$$

Such a square is given precisely by $\mathsf{ap}_{\widetilde{p}_n}(\mathsf{push}^{(2)}(a_0, \dots, a_{n-1}))$, and we are done. □

Let us restate the result in a more compact and self-contained form.

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

Corollary 31. *Let $f, g : \bigwedge_{i\leq n} A_i \to B$. If f and g agree on $\iota : \widetilde{\bigwedge}_{i\leq n} A_i \to \bigwedge_{i\leq n} A_i$, that is if $f \circ \iota = g \circ \iota$, then $f = g$.*

For completeness, let us also state the analogous result for pointed maps.

Corollary 32. *Let $f, g : \bigwedge_{i\leq n} A_i \to_\star B$. If $f \circ \iota = g \circ \iota$ as pointed map, then we obtain an equality of pointed maps $f = g$.*

7. Conclusions and Future Work

Let us summarise the key contributions of this paper. First and foremost, we have shown in Sections 3 through 5 that the smash product forms a 1-coherent symmetric monoidal product on the wild category of pointed types (Theorem 24). This fills a long-standing and rather troublesome gap in the HoTT literature. In particular, it implies the correctness of previous work which relies on this result – perhaps most notably, that of Brunerie (2016) and van Doorn (2018).

Second, we have presented a new method for reasoning about smash products in HoTT. This was first done by introducing, in Section 4, an informal heuristic which was used heavily leading up to Theorem 24. We later provided an attempt at a formal version of the heuristic in Corollaries 31 and 32. The message of these results is simple: functions defined over smash products in HoTT are not much harder to deal with than those defined over ordinary Cartesian products. This contradicts a commonly held view that smash products in HoTT become unworkable in higher dimensions.

In addition to salvaging already existing work relying on unproved results about smash products, we also hope that the results presented here will be useful in the further development of synthetic homotopy theory in HoTT. In particular, Theorem 24 is not the only consequence of Propositions 25 and 26. These propositions capture two important properties of the equivalence $\wedge_{n,m} : S^n \wedge S^m \simeq S^{n+m}$. For instance, the special case when $n = m = 1$ was implicitly used by Ljungström and Mörtberg (2023, Section VI.) in order to provide a simplified version of Brunerie's proof of $\pi_4(S^3) \cong \mathbb{Z}/2\mathbb{Z}$. More generally, $\wedge_{n,m} : S^n \wedge S^m \simeq S^{n+m}$ gives rise to the *Whitehead product* (Whitehead, 1941). This operation was originally defined in HoTT by Brunerie (2016, Definition 3.3.3.) using joins of spheres, but we can also give it a rather compact construction in terms of $\wedge_{n,m}$: the Whitehead product $[f, g]$ of two maps $f : S^{n+1} \to_\star A$ and $g : S^{m+1} \to_\star A$ can be viewed as the map defined by the composition:

$$S^{n+m} \xrightarrow{\wedge_{n,m}^{-1}} (S^n \wedge S^m) \xrightarrow{\mathsf{comm}} \Omega(S^{n+1} \vee S^{m+1}) \xrightarrow{\Omega(f\vee g)} \Omega A$$

where comm sends $\langle x, y\rangle$ to $\sigma_l^{-1}(x) \cdot \sigma_r(y) \cdot \sigma_l(x) \cdot \sigma_r^{-1}(y)$, where $\sigma : S^k \to \Omega S^{k+1}$ is defined by $\sigma(x) = \mathsf{merid}(x) \cdot \mathsf{merid}(\mathsf{north})^{-1}$ and the subscripts l and r indicate in which component of $S^{n+1} \vee S^{m+1}$ the loop takes place. This induces a map on homotopy groups

$$\pi_{n+1}(A) \times \pi_{m+1}(A) \to \pi_{n+m}(\Omega(A)) \xrightarrow{\sim} \pi_{n+m+1}(A)$$

which, classically, turns $\pi_\bullet(A)$ into a graded quasi-Lie algebra (Uehara and Massey 1957). This fact is still open in HoTT, but we conjecture that Propositions 25 and 26 will play a crucial role in a prospective proof. More concretely, as $\wedge_{n,m}$ is used in the construction of this map, Proposition 25 and 26 should, respectively, play important roles in prospective proofs of graded symmetry and the Jacobi identity.

On a related note, the heuristic presented in Section 4 and the related results in Section 6 are reminiscent of results from the study of polyhedral products (see e.g., Bahri et al. 2019; Theriault 2018). This is a field of mathematics which, very coarsely speaking, explores and generalises the mediation between iterated wedge sums and iterated Cartesian products. It is unclear whether

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

this topic can be studied in a meaningful way through the lens of HoTT, but exploring this would certainly amount to an interesting continuation of the work we have presented here.

Another interesting question which we have not covered here is whether our heuristic can be used to show the primary technical gap in the thesis of van Doorn (2018), namely that the equivalence (1) is a pointed natural equivalence. This would show that the universe of pointed types is a closed monoidal (wild) category. It is not obvious that the heuristic, as it is phrased in this paper, is suitable for this problem. Nevertheless, it is not unlikely that techniques similar to those used here still apply. We leave this for future work.

Acknowledgements. The author would like to thank Evan Cavallo for several fruitful discussions, for his prior unpublished formalisations of smash products in Cubical Agda which have been useful as a source of inspiration for the formalisation of this paper, and, of course, for his contribution of the incredibly useful Lemma 14 to the homotopy type theory literature. The author would also like to thank Dan Petersen and an anonymous reviewer for their insightful comments on earlier versions of this paper.

Funding statement. This paper is based on research supported by the Swedish Research Council (Vetenskapsrådet) under Grant No. 2019-04545.

Competing interests. The author declares none.

Note

1 By 'computes to' we do not mean 'normalises in Agda to'. We mean 'compute' in the manual sense, that is, 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

Bahri, A., Bendersky, M. and Cohen, F. R. (2019). Polyhedral products and features of their homotopy theory. In Miller, H. (ed.) *Handbook of Homotopy Theory* (1st ed.). New York: CRC Press, pp. 103–144.

Brunerie, G. (2016). *On the Homotopy Groups of Spheres in Homotopy Type Theory*. PhD thesis, Université Nice Sophia Antipolis.

Brunerie, G. (2018). Computer-generated proofs for the monoidal structure of the smash product. *Homotopy Type Theory Electronic Seminar Talks*.

Brunerie, G., Ljungström, A. and Mörtberg, A. (2022). Synthetic integral cohomology in cubical agda. In Manea, F. and Simpson, A.(eds.) *30th EACSL Annual Conference on Computer Science Logic (CSL 2022), Dagstuhl, Germany*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, vol. **216**, 11:1–11:19, *Leibniz International Proceedings in Informatics (LIPIcs)*.

Buchholtz, U., Christensen, J. D., Flaten, J. G. T. and Rijke, E. (2023). Central h-spaces and banded types, arXiv: 2301.02636.

Capriotti, P. and Kraus, N. (2017). Univalent higher categories via complete semi-segal types. *Proceedings of the ACM on Programming Languages* **2** (POPL) 44:1–44:29.

Cavallo, E. (2021a). *Higher Inductive Types and Internal Parametricity for Cubical Type Theory*. Phd thesis, Carnegie Mellon University.

Cavallo, E. (2021b). Pointed functions into a homogeneous type are equal as soon as they are equal as unpointed functions. Agda formalization, part of the cubical library. Available at https://agda.github.io/cubical/Cubical. Foundations.Pointed.Homogeneous.html#1616.

Cavallo, E. and Harper, R. (2020). Internal parametricity for cubical type theory. In Fernández, M. and Muscholl, A.(eds.) *Leibniz International Proceedings in Informatics (LIPIcs). 28th EACSL Annual Conference on Computer Science Logic (CSL 2020), Dagstuhl, Germany*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, vol. **152**, 13:1–13:17.

Kraus, N. (2013). The Truncation Map $|\_|:N\to\|N\|$ is nearly Invertible. Blog post at https://homotopytypetheory.org/2013/10/28/

Lamiaux, T., Ljungström, A. and Mörtberg, A. (2023). Computing cohomology rings in Cubical Agda. In: *Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2023, New York, NY, USA*. Association for Computing Machinery, 239–252.

https://doi.org/10.1017/S0960129524000318 Published online by Cambridge University Press

Ljungström, A. and Mörtberg, A. (2023). Formalizing $\pi_4(S^3) \cong \mathbb{Z}/2\mathbb{Z}$ and computing a Brunerie number in Cubical Agda. In: *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), Los Alamitos, CA, USA*. IEEE Computer Society, 1–13.

Ljungström, A. and Mörtberg, A. (2024). Computational synthetic cohomology theory in homotopy type theory, arXiv: 2401.16336.

Pujet, L. and Mörtberg, A. (2020). Cubical synthetic homotopy theory. In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New York, NY, USA*. Association for Computing Machinery, 158–171.

The Univalent Foundations Program. (2013). Homotopy Type Theory: Univalent Foundations of Mathematics. Self-published, Institute for Advanced Study.

Theriault, S. (2018). The dual polyhedral product, cocategory and nilpotence. *Advances in Mathematics* **340** 138–192.

Uehara, H. and Massey, W. S. (1957). *The Jacobi Identity for Whitehead Products*, Princeton, Princeton University Press, 361–377.

van Doorn, F. (2018). *On the Formalization of Higher Inductive Types and Synthetic Homotopy Theory*. PhD thesis, Carnegie Mellon University.

Vezzosi, A., Mörtberg, A. and Abel, A. (2021). Cubical agda: a dependently typed programming language with univalence and higher inductive types. *Journal of Functional Programming* **31** e8.

Whitehead, J. H. C. (1941). On adding relations to homotopy groups. *Annals of Mathematics* **42** (2) 409–428.

Cite this article: Ljungström A (2024). Symmetric monoidal smash products in homotopy type theory. *Mathematical Structures in Computer Science* **34**, 985–1007. https://doi.org/10.1017/S0960129524000318