1-Types versus Groupoids

Niels van der Weide, Dan Frumin, and Herman Geuvers

Radboud Univeristy Nijmegen, The Netherlands
nweide@cs.ru.nl, dfrumin@cs.ru.nl, herman@cs.ru.nl

In homotopy type theory, a set is a type A such that for each $x, y : A$ and $p, q : x = y$ we have $p = q$ [9]. In other words, equality of A is proof-irrelevant. A setoid is a type A together with an equivalence relation on A. From a set A we can construct a setoid, namely A with the relation $\lambda xy. x = y$. Conversely, from a setoid we can construct a set with quotient types. For a type A and an equivalence relation $R : A \to A \to \text{hProp}$, the set quotient A/R can be constructed as a higher inductive type with a point constructor $\text{cl} : A \to A/R$, a path constructor $\text{eqcl} : \prod_{x,y:A} R\ x\ y \to \text{cl}\ x = \text{cl}\ y$, and one saying that A/R is a set [7].

Now let us go to dimension 1. A 1-type is a type A such that for each $x, y : A$ the type $x = y$ is a set. The 1-dimensional version of setoids is groupoids. A groupoid on A consists of a family $G : A \to A \to \text{hSet}$, an identity $e : \prod_{x:A} G\ x\ x$, and operations $()^{-1} : \prod_{x,y:A} G\ x\ y \to G\ y\ x$ and $(\cdot) : \prod_{x,y,z:A} G\ x\ y \to G\ y\ z \to G\ x\ z$ satisfying the usual laws for associativity, neutrality, and inverses. We write $\text{grpd}\ A$ for the type of groupoids on A.

Every 1-type A gives rise to a path groupoid $P\ A$ on A defined by $\lambda xy. x = y$. However, can we go the other direction? More precisely, given a groupoid G on a type A, our goal is to construct a 1-type $\text{gquot}\ A\ G$ together with a map $\text{gcl} : A \to \text{gquot}\ A\ G$ such that for each $x, y : A$ the types $\text{gcl}\ x = \text{gcl}\ y$ and $G\ x\ y$ are equivalent. To do so, we use the following higher inductive type, which we call the *groupoid quotient*.

Higher Inductive Type $\text{gquot}\ (A : \text{Type})\ (G : \text{grpd}\ A) :=$

$\mid \text{gcl} : A \to \text{gquot}\ A\ G$

$\mid \text{gcleq} : \prod_{x,y:A} G\ x\ y \to \text{gcl}\ x = \text{gcl}\ y$

$\mid \text{ge} : \prod_{x:A} \text{gcleq}\ x\ (e\ x) = \mathbf{refl}$

$\mid \text{ginv} : \prod_{x,y:A} \prod_{g:G\ x\ y} \text{gcleq}\ y\ x\ (g^{-1}) = (\text{gcleq}\ x\ y\ g)^{-1}$

$\mid \text{gconcat} : \prod_{x,y,z:A} \prod_{g_1:G\ x\ y} \prod_{g_2:G\ y\ z} \text{gcleq}\ x\ z\ (g_1 \cdot g_2) = \text{gcleq}\ x\ y\ g_1\ @\ \text{gcleq}\ y\ z\ g_2$

$\mid \text{gtrunc} : \prod_{x,y:\text{gquot}\ A\ G} \prod_{p,q:x=y} \prod_{r,s:p=q} r = s$

To derive the elimination and computation rules of this type, we use the method by Dybjer and Moenclaey [4]. The equations ge, ginv, and gconcat show that gcleq is a homomorphism of groupoids. The constructor gtrunc guarantees that $\text{gquot}\ A\ G$ is a 1-type.

For quotients, the types $\text{cl}\ x = \text{cl}\ y$ and $R\ x\ y$ are equivalent [7]. For $\text{gquot}\ A\ G$, we have a similar statement.

Proposition. *For every $x, y : A$ the types $\text{gcl}\ x = \text{gcl}\ y$ and $G\ x\ y$ are equivalent.*

Each equivalence relation induces a groupoid. The quotient of such an induced groupoid coincides with the set quotient.

Proposition. *If A is a type and R is an equivalence relation on A, then $A/R \simeq \text{gquot}\ A\ \overline{R}$ where $\overline{R}$ is the groupoid induced by R.*

In addition, every 1-type is the groupoid quotient of its path groupoid.

Proposition. *For all 1-types A, we have $A \simeq \text{gquot}\ A\ (P\ A)$ with $P\ A$ the path groupoid on A.*

The category of groupoids has products and coproducts. More precisely, for groupoids $G_1 : \text{grpd}\ A$ and $G_2 : \text{grpd}\ B$, we define $G_1 \times G_2 : \text{grpd}\ (A \times B)$ where the elements are pairs of G_1 and G_2 and the operations are defined pointwise. Similarly, we define $G_1 + G_2 : \text{grpd}\ (A + B)$.

Proposition. *The* gquot *construction commutes with sums and products. More precisely,*

$$\mathrm{gquot}\ (A \times B)\ (G_1 \times G_2) \simeq \mathrm{gquot}\ A\ G_1 \times \mathrm{gquot}\ B\ G_2,$$

$$\mathrm{gquot}\ (A + B)\ (G_1 + G_2) \simeq \mathrm{gquot}\ A\ G_1 + \mathrm{gquot}\ B\ G_2.$$

The proofs of these propositions have been formalized in Coq using the HoTT library [2] and they are available at `https://github.com/nmvdw/groupoids`.

Conclusion and Further Work. Groupoids form a model of type theory [5, 8]. Since 1-types are preserved under dependent products, sums, and identity types [9], they also form a model. This work gives a partial internal comparison between these models.

Dybjer and Moenclaey give an interpretation of higher inductive types in the groupoid model [4]. Internalizing their construction and applying gquot gives an interpretation of 1-HITs as 1-types. The first proposition then characterizes $||x = y||_0$ for this interpretation of higher inductive types.

Another interesting generalization would be using higher groupoids instead of plain groupoids. One can develop a theory of higher groupoids in HoTT similar to the theory of higher groups [3]. This could give a comparison between n-types and n-groupoids. In addition, a full version would be a comparison between ω-groupoids and types [1, 6].

Acknowledgements. We thank Joshua Moerman for giving the inspiration for the last proposition.

References

[1] Thorsten Altenkirch and Ondrej Rypacek. A Syntactical Approach to Weak ω-Groupoids. In *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012*, pages 16–30, 2012.

[2] Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Michael Shulman, Matthieu Sozeau, and Bas Spitters. The HoTT Library: A Formalization of Homotopy Type Theory in Coq. In *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs*, CPP 2017, pages 164–172, New York, NY, USA, 2017. ACM.

[3] Ulrik Buchholtz, Floris van Doorn, and Egbert Rijke. Higher Groups in Homotopy Type Theory. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2018.

[4] Peter Dybjer and Hugo Moeneclaey. Finitary Higher Inductive Types in the Groupoid Model. In *Proceedings of MFPS*, 2017.

[5] Martin Hofmann and Thomas Streicher. The Groupoid Interpretation of Type Theory. *Twenty-five years of constructive type theory (Venice, 1995)*, 36:83–111, 1998.

[6] Peter LeFanu Lumsdaine. Weak ω-Categories from Intensional Type Theory. In Pierre-Louis Curien, editor, *Typed Lambda Calculi and Applications*, pages 172–187, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[7] Egbert Rijke and Bas Spitters. Sets in Homotopy Type Theory. *Mathematical Structures in Computer Science*, 25(5):1172–1202, 2015.

[8] Matthieu Sozeau and Nicolas Tabareau. Towards an Internalization of the Groupoid Model of Type Theory. In *Types for Proofs and Programs 20th Meething (TYPES 2014), Book of Abstracts*, 2014.

[9] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.