An elementary definition of globular weak ∞-groupoids

Guillaume Brunerie

Institute for Advanced Study

CMS Summer Meeting, Halifax, NS, Canada
June 5, 2013

Why yet another definition of (weak) ∞-groupoids?

- It's equivalent to Grothendieck's definition (up to some minor changes)
- It's short and simple
- It's an essentially algebraic definition (i.e. an ∞-groupoid is a globular set equipped with a collection of operations)

Why yet another definition of (weak) ∞-groupoids?

- It's equivalent to Grothendieck's definition (up to some minor changes)
- It's short and simple
- It's an essentially algebraic definition (i.e. an ∞-groupoid is a globular set equipped with a collection of operations)

Why yet another definition of (weak) ∞-groupoids?

- It's equivalent to Grothendieck's definition (up to some minor changes)
- It's short and simple
- It's an essentially algebraic definition (i.e. an ∞-groupoid is a globular set equipped with a collection of operations)

Why yet another definition of (weak) ∞-groupoids?

- It's equivalent to Grothendieck's definition (up to some minor changes)
- It's short and simple
- It's an essentially algebraic definition (i.e. an ∞-groupoid is a globular set equipped with a collection of operations)

Outline

Globular sets

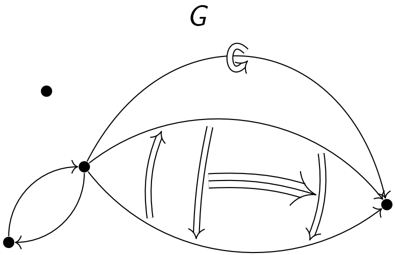
Proposition

A *globular set* G *has*

- *a set of* *objects* $\mathrm{Ob}(G)$

$$a : G \overset{def}{\iff} a \in \mathrm{Ob}(G)$$

- *for all* $a, b : G$*, a globular set of* *morphisms* $\mathrm{Hom}_G(a, b)$



Globular sets

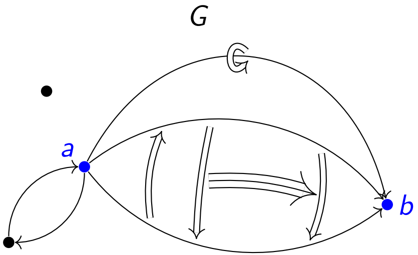
Proposition

A globular set G has

- *a set of objects* $\mathrm{Ob}(G)$

$$a : G \overset{def}{\iff} a \in \mathrm{Ob}(G)$$

- *for all* $a, b : G$*, a globular set of morphisms* $\mathrm{Hom}_G(a, b)$



Globular sets

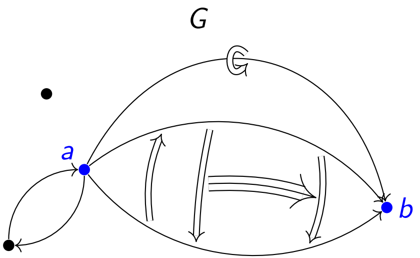
Proposition

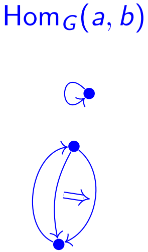
A globular set G has

- *a set of objects* Ob(G)

$$a : G \overset{def}{\iff} a \in \mathrm{Ob}(G)$$

- *for all $a, b : G$, a globular set of morphisms* $\mathrm{Hom}_G(a, b)$





Bigroupoids

Definition

A bigroupoid is a globular set G with no n-cells for $n \geq 3$ and equipped with the following 13 operations (and some equations):

Bigroupoids

Definition

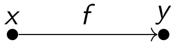
A bigroupoid is a globular set G with no n-cells for $n \geq 3$ and equipped with the following 13 operations (and some equations):

$\overset{x}{\bullet}$	$\mapsto$	$1_x : \mathrm{Hom}_G(x, x)$
$x \xrightarrow{f} y$	$\mapsto$	$f^{-1} : \mathrm{Hom}_G(y, x)$
$x \xrightarrow{f} y \xrightarrow{g} z$	$\mapsto$	$g \circ f : \mathrm{Hom}_G(x, z)$
$x \xrightarrow{f} y \xrightarrow{g} z \xrightarrow{h} d$	$\mapsto$	$\alpha_{f,g,h} : \mathrm{Hom}_{\dots}(h \circ (g \circ f), (h \circ g) \circ f)$
$x \xrightarrow{f} y$	$\mapsto$	$\lambda_f : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(1_y \circ f, f)$
$x \xrightarrow{f} y$	$\mapsto$	$\rho_f : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(f \circ 1_x, f)$

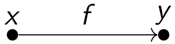
Bigroupoids

Definition

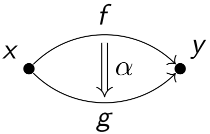
A bigroupoid is a globular set G with no n-cells for $n \geq 3$ and equipped with the following 13 operations (and some equations):



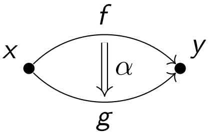
$$\mapsto \quad \lambda'_f : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(f^{-1} \circ f, 1_x)$$



$$\mapsto \quad \rho'_f : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(f \circ f^{-1}, 1_y)$$



$$\mapsto \quad i(\alpha) : \mathrm{Hom}_{\mathrm{Hom}_G(y,x)}(f^{-1}, g^{-1})$$

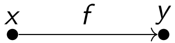


$$\mapsto \quad \widetilde{\alpha^{-1}} : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(g, f)$$

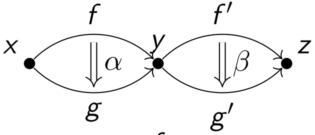
Bigroupoids

Definition

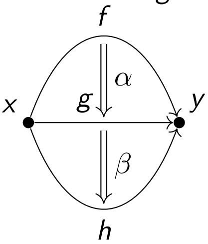
A **bigroupoid** is a globular set G with no n-cells for $n \geq 3$ and equipped with the following 13 operations (and some equations):



$$\mapsto \quad 1_f : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(f, f)$$

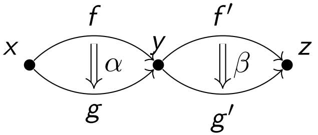


$$\mapsto \quad \beta \star_0 \alpha : \mathrm{Hom}_{\mathrm{Hom}_G(x,z)}(f' \circ f, g' \circ g)$$



$$\mapsto \quad \beta \star_1 \alpha : \mathrm{Hom}_{\mathrm{Hom}_G(x,y)}(f, h)$$

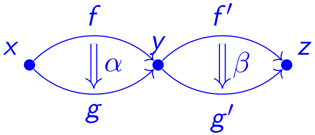
Anatomy of an operation



$\mapsto \; \beta \star_0 \alpha : \mathrm{Hom}_{\mathrm{Hom}_G(x,z)}(f' \circ f, g' \circ g)$

Every operation has two parts:

Anatomy of an operation

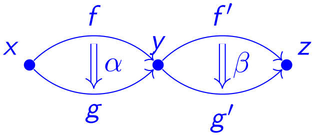


$\mapsto \beta \star_0 \alpha : \mathrm{Hom}_{\mathrm{Hom}_G(x,z)}(f' \circ f, g' \circ g)$

Every operation has two parts:

- The input: is a bunch of variables with a "contractible" shape

Anatomy of an operation

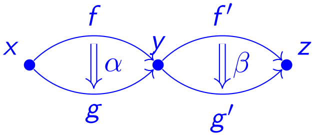


$\mapsto \quad \beta \star_0 \alpha : \mathrm{Hom}_{\mathrm{Hom}_G(x,z)}(f' \circ f, g' \circ g)$

Every operation has two parts:

- The input: is a bunch of variables with a “contractible” shape
- The output: is an iterated hom globular set of G described using only
  - the variables in the input
  - possibly some other operations of the structure of ∞-groupoid

Anatomy of an operation



$\mapsto \quad \beta \star_0 \alpha : \mathrm{Hom}_{\mathrm{Hom}_G(x,z)}(f' \circ f, g' \circ g)$

Every operation has two parts:

- The input: is a bunch of variables with a "contractible" shape
- The output: is an iterated hom globular set of G described using only
  - the variables in the input
  - possibly some other operations of the structure of ∞-groupoid

Conversely, any contractible input and well-formed output should correspond to some operation of ∞-groupoids.

Idea

Define a *syntax* in which we can express the notion of

- contractible input
- well-formed output

Idea

Define a *syntax* in which we can express the notion of

- contractible input
- well-formed output

Two basic syntactic notions:

- types
  $\rightarrow$ represent iterated hom globular sets of G
- terms
  $\rightarrow$ represent cells of G

Every term has an associated type. We write:

$$u : A \overset{\text{def}}{\iff} \text{the term } u \text{ has type } A$$

Contexts

In a term like $g \circ f$, we need to know what f and g are.
$\rightarrow$ types and terms are considered in a given *context*

context = list of variables you are allowed to use

Definition

A context is a sequence $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$ of variables and types such that:

- the variables x_i are pairwise distinct
- for every $k \in \{1, \ldots, n\}$, A_k is a type in the context $(x_1 : A_1, \ldots, x_{k-1} : A_{k-1})$

Types

Definition

In a context Γ, a type is either:

- the base type (which corresponds to G):

$$\star$$

- a hom type (which corresponds to $\mathrm{Hom}_A(u, v)$):

$$u \simeq_A v$$

In the last case we require that

- A is a type in Γ
- u and v are terms of type A in Γ

Context and types (examples)

Examples (well-formed contexts)

$$(x : \star), (y : \star), (z : \star), (f : x \simeq_\star y), (g : y \simeq_\star z)$$

and

$$(f : \star), (g : \star)$$

Counterexamples (ill-formed contexts)

$$(f : x \simeq_\star y), (g : y \simeq_\star z)$$

and

$$(x : \star), (y : \star), (f : x \simeq_\star y), (g : y \simeq_\star x), (\alpha : f \simeq_? g)$$

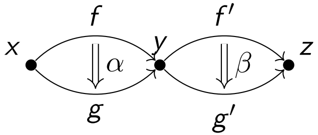
Contractible contexts

Definition

A context Δ is contractible if either:

- Δ is a singleton
$$\Delta = (x : \star)$$
- Δ is obtained from some contractible context Δ' by duplicating a variable $(x : A) \in \Delta'$ and "gluing a ball":
$$\Delta = (\Delta', (y : A), (z : x \simeq_A y))$$
(where y and z are fresh variables)

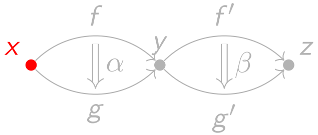
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x : \star), (y : \star), (f : x \simeq_\star y), (g : x \simeq_\star y), (\alpha : f \simeq_{x \simeq_\star y} g),$$
$$(z : \star), (f' : y \simeq_\star z), (g' : y \simeq_\star z), (\beta : f' \simeq_{y \simeq_\star z} g'))$$

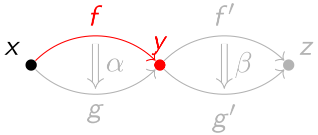
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x : \star), (y : \star), (f : x \simeq_\star y), (g : x \simeq_\star y), (\alpha : f \simeq_{x \simeq_\star y} g), (z : \star), (f' : y \simeq_\star z), (g' : y \simeq_\star z), (\beta : f' \simeq_{y \simeq_\star z} g'))$$

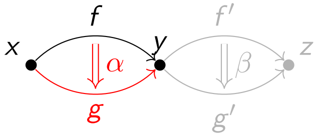
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x : \star), (y : \star), (f : x \simeq_\star y), (g : x \simeq_\star y), (\alpha : f \simeq_{x \simeq_\star y} g),$$
$$(z : \star), (f' : y \simeq_\star z), (g' : y \simeq_\star z), (\beta : f' \simeq_{y \simeq_\star z} g'))$$

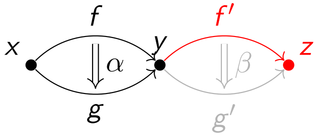
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x : \star), (y : \star), (f : x \simeq_\star y), (g : x \simeq_\star y), (\alpha : f \simeq_{x \simeq_\star y} g),$$
$$(z : \star), (f' : y \simeq_\star z), (g' : y \simeq_\star z), (\beta : f' \simeq_{y \simeq_\star z} g'))$$

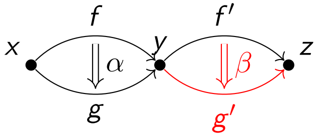
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x : \star), (y : \star), (f : x \simeq_\star y), (g : x \simeq_\star y), (\alpha : f \simeq_{x \simeq_\star y} g),$$
$$(z : \star), (f' : y \simeq_\star z), (g' : y \simeq_\star z), (\beta : f' \simeq_{y \simeq_\star z} g'))$$

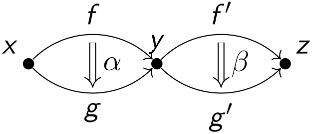
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x : \star), (y : \star), (f : x \simeq_\star y), (g : x \simeq_\star y), (\alpha : f \simeq_{x \simeq_\star y} g), (z : \star), (f' : y \simeq_\star z), (g' : y \simeq_\star z), (\beta : f' \simeq_{y \simeq_\star z} g'))$$

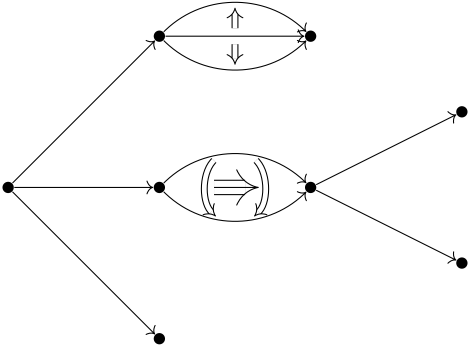
Contractible contexts (example)



corresponds to

$$\Delta_{\star_0} := ((x:\star),(y:\star),(f : x \simeq_\star y),(g : x \simeq_\star y),(\alpha : f \simeq_{x \simeq_\star y} g),$$
$$(z:\star),(f' : y \simeq_\star z),(g' : y \simeq_\star z),(\beta : f' \simeq_{y \simeq_\star z} g'))$$

Contractible contexts (another example)



Terms

Definition

In a context Γ, a term is either:

- a variable x such that $(x : A) \in \Gamma$ for some A
- a coherence cell

$$\mathrm{coh}_{\Delta.A}(u_1, \ldots, u_n)$$

For a coherence cell we require that:

- the input $\Delta = (x_1 : B_1, \ldots, x_n : B_n)$ is a contractible context
- the output A is a type in Δ
- in the context Γ, the arguments u_k are terms satisfying

$$u_k : B_k[x_1 := u_1, \ldots, x_{k-1} := u_{k-1}]$$

The type of $\mathrm{coh}_{\Delta.A}(u_1, \ldots, u_n)$ is $A[x_1 := u_1, \ldots, x_n := u_n]$.

Terms (example)

Example (Identity)

If $u : \star$ in context Γ, then

$$\mathsf{coh}_{(x:\star).(x \simeq_\star x)}(u) : u \simeq_\star u$$

∞-groupoids

Definition

An ∞-groupoid is a globular set G equipped with, for every contractible context Δ and every type A in Δ, an operation

$$\mathbf{coh}_{\Delta.A} : (\eta \in [\![\Delta]\!]) \to \mathrm{Ob}([\![A]\!]_\eta)$$

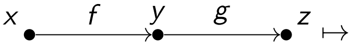
where

$$\begin{aligned}
[\![\star]\!]_\eta &:= G \\
[\![u \simeq_A v]\!]_\eta &:= \mathrm{Hom}_{[\![A]\!]_\eta}([\![u]\!]_\eta, [\![v]\!]_\eta) \\
[\![x]\!]_\eta &:= \eta(x) \\
[\![\mathrm{coh}_{\Delta.A}(u_1, \ldots, u_n)]\!]_\eta &:= \mathbf{coh}_{\Delta.A}([\![u_1]\!]_\eta, \ldots, [\![u_n]\!]_\eta) \\
[\![(x_1 : B_1, \ldots, x_n : B_n)]\!] &:= \{(a_1, \ldots, a_n) \mid a_k \in \mathrm{Ob}([\![B_k]\!]_{(a_1,\ldots,a_{k-1})})\}
\end{aligned}$$

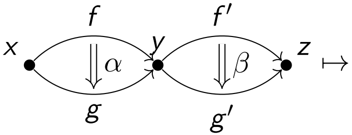
Examples of operations

Example

Every operation of bigroupoids corresponds to some $\mathbf{coh}_{\Delta.A}$.



$$g \circ f : x \simeq_{\star} z$$



$$\beta \star_0 \alpha : (f' \circ f) \simeq_{(x \simeq_{\star} z)} (g' \circ g)$$

More generally, everything with a contractible input and a well-formed output.

Examples of ∞-groupoids

Examples

- Groupoids
- The fundamental ∞-groupoid of a topological space

∞-groupoid of morphisms

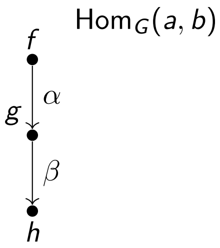
Proposition

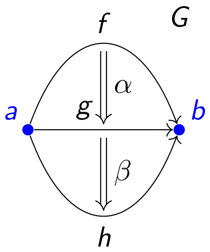
Let G be an ∞-groupoid and $a, b : G$.
Then $\mathrm{Hom}_G(a, b)$ has a natural structure of ∞-groupoid.

∞-groupoid of morphisms

Proposition

Let G be an ∞-groupoid and $a, b : G$.
Then $\mathrm{Hom}_G(a,b)$ has a natural structure of ∞-groupoid.





$$\mathbf{coh}^{\mathrm{Hom}_G(a,b)}_{\Delta.A}(u_1,\ldots,u_n) := \mathbf{coh}^{G}_{\Sigma\Delta.\Sigma A}(a,b,u_1,\ldots,u_n)$$

Summary

We defined *syntactically* the notions of

- contractible input
- well-formed output according to a given input
- coherence operation

This gives an elementary definition of globular weak ∞-groupoid which is

- easy to define and understand
- equivalent to known definitions
- maybe easier to work with

Summary

We defined *syntactically* the notions of

- contractible input
- well-formed output according to a given input
- coherence operation

This gives an elementary definition of globular weak ∞-groupoid which is

- easy to define and understand
- equivalent to known definitions
- maybe easier to work with

Summary

We defined *syntactically* the notions of

- contractible input
- well-formed output according to a given input
- coherence operation

This gives an elementary definition of globular weak ∞-groupoid which is

- easy to define and understand
- equivalent to known definitions
- maybe easier to work with

Thank you for your attention