ARTIFICIAL CO-CREATIVE GENERATION OF THE NOTION OF TOPOLOGICAL GROUP BASED ON CATEGORICAL CONCEPTUAL BLENDING

DANNY ARLEN DE JESÚS GÓMEZ-RAMÍREZ, YOE A. HERRERA-JARAMILLO, AND FLORIAN GEISMANN

Abstract. We use a computationally-feasible formalization of conceptual blending as colimits of many-sorted first-order theories for presenting a concrete formal generation of the notion of topological group. This represents a genuine extension of current work done in this direction.

Artificial Concept Invention based on the seminal cognitive mechanism of conceptual blending has obtained a central place within intelligent and creative systems' research. We provide highly abstract extensions of work done in this direction for the fundamental mathematical notion of topological groups. Specifically, we start with two basic notions belonging to topology and abstract algebra, and we describe recursively formal specifications in the Common Algebraic Specification Language (CASL). The notion of conceptual blending between such conceptual spaces can be materialized computationally in the Heterogeneous Tool Set (HETS). The fundamental notion of topological groups is explicitly generated through three different artificial specifications based on conceptual blending and conceptual identification, starting with the concepts of continuous functions and mathematical groups (described with minimal set-theoretical conditions).

Keywords: Artificial conceptual creation; Concept invention; Formal conceptual blending; Conceptual identification; Colimits; Topological groups.

1. Introduction

Latest advances in computational creativity, cognitive and computer science continue enhancing our understanding about the way in which our minds create mathematics at high levels of sophistication [6]. In particular, more precise formalization of fundamental cognitive mechanisms for conceptual creation has been developed and tested in several mathematical domains [14, 10, 18, 11]. Among those basic cognitive abilities conceptual blending has shown to be not only one of the most powerful, but also one of the most omnipresent among mathematics [1, 7]. For instance, seminal notions of (algebraic) number

1991 *Mathematics Subject Classification.* 22-04, 68T99 .

theory, Fields and Galois theory and commutative algebra have been conceptually meta-generated (with a computational basis) in terms of a categorical formalization of conceptual blending [10, 4, 11, 3].

More concretely, the blend of a 'V'-shaped diagram between two input mathematical concepts with a generic (base) concept is characterized by means of categorical colimits. Now, such a colimit exists because mathematical concepts are formalized in terms of many-sorted first-order theories with axiom-preserving morphisms [10, §8.2]. Another basic metamathematical cognitive mechanism commonly used during mathematical research is *conceptual identification*; i.e., the ability of (cognitively) interpreting two (abstract) concepts as the same, with the purpose of simplifying inferential processes on the mind [12, 13].

Research in this direction is also closely related with the development of new forms of cognitively-inspired artificial intelligence on the domain of abstract mathematical discovery/creation [10]., specifically, within the multidisciplinary research program Artificial Mathematical Intelligence [10, §8.7.1].[1]

In the modern literature concerning formal (artificial) conceptual generation based on conceptual blending and metaphorical reasoning et al., an special attention has been set to the study of algebraic and arithmetic notions. On the other hand, concepts with a more topological and geometrical nature represents a relatively unexplored field in this regard. So, we aim to start to fill this gap presenting a detailed conceptual generation of the seminal concept of topological group starting with the elementary notions of group, continuous function (between topological spaces) and 'perfect square' topological space. We will present the corresponding pseudo-specifications described with the common algebraic specification language (CASL) [2] and implicitly using the formalisms described by the Heterogeneous Tool Set (HETS) [15]. HETS is a suitable software because it provides specific tools for computing formal colimits for the above formalization of concepts described above.

2. Conceptual Preliminaries

For the sake of completeness in the presentation, we recall the initial notions that we will use as foundational bricks of our conceptual 'building'.

First of all, a group $(G, +, e)$ is simply a set equipped with a binary operation $+$ and an outstanding element $e \in G$, such that e is the neutral element with respect to $+$, $+$ is associative and each element possesses a inverse.

Second, a topological space (X, T) consists of a set X and a collection T of subsets of X satisfying the following conditions: $\emptyset, X \in T$, T is closed under finite intersections and arbitrary unions. So, a function $f : X \to Y$ between topological spaces (X, T_X) and (Y, T_Y) is continuous if and only if for any

[1] www.ArtificialMathematicalIntelligence.com

$U \in T_Y$, $f^{-1}(U) \in T_X$. In the case that X and Y are exactly the same topological space, f is alternatively called a continuous endomorphism.

Third, a topological space (Z, T_Z) is called a perfect square topological space if and only if there exists another topological space (X, T_X) such that $Z = X \times X$ and T_Z is exactly the product topology consisting of arbitrary unions of finite intersections of Cartesian products of elements of T_x (all viewed embedded in Z).

Fourth, a continuous binary operation (over a topological space (X, T) is a continuous function $\oplus : X \times X \to X$, where $X \times X$ is assumed to have the product topology.

Fifth, a topological group $(G, +, e)$ is a group which is at the same time a topological space such that the operations $+$ and $Addinv : G \to G$ (sending $x \to -x$) are continuous functions. If the continuity of the inverse function is not required, then $(G, +, e)$ is a pseudo topological group.

For a more detailed reading of the former concepts the interested reader may consult [9] and [16].

3. Conceptual Generation of the Notion of Topological Group in terms of Formal Conceptual Blending and Metaphorical Reasoning

Due to the fact that we want to construct artificial specifications of mathematical notions ‘from scratch’, we will describe along with the central axioms of each of the concepts, the minimal set-theoretical information needed to be able to do robust conceptual operations with them. Moreover, we present all the pseudo-codes in the most natural and clear way possible, so that working mathematicians with little experience with CASL would understand the essentials features of the mathematical structures involved.

3.1. **Continuous Binary Operation.**

In the following specifications we will generate the notion of continuous binary operation as the formal blend between the notions of continuous functions (between topological spaces) and perfect square topological space. We use extra constants for some sorts, denoted with an additional ‘prime’ symbol (e.g. A'), due to the fact that we need to be able to manipulate each sort as a ‘set’ as well. Similarly, we will define a new constant for ‘simulating’ the Cartesian product of a set with itselt, because CASL do not deal with Cartesian products between sorts as constants. The importance of this technical trick can be better appreciated after reading completely each of the specifications. Due to the fact that we are showing different specifications and for the sake of simplicity in the presentation we use ellipsis (‘$\cdots$’) in the pseudo-code to indicate that already defined concepts should be specified again. Now, such technicalities did not appear so explicitly in daily mathematical research because our minds do conceptual identifications almost automatically.

```
begin CASL

%%  Continuous function between two topological spaces

spec ContFunc =
     sorts   Sets, A, TA, PA, B, TB, PB;
            A, TA, PA, B, TB, PB < Sets;
            TA < PA, TB < PB
%%  A = domain of the function, TA = topology of A, PA =
    powerset of A
%%  B = codomain of B, TB = topology of B, PB  = powerset of B
      ops   EmpSet, A', TA', PA', B', TB', PB' : Sets;
            __ in __ : Sets × Sets
            __ inter __ : Sets × Sets ↦ Sets
            Uni__ : Sets
            __ subset __ : Sets × Sets
            f: A ↦ B
            inversef: TB ↦ TA
%%  Definition of A, TA and PA
      preds ∀ a : Sets. a : A ⇔ a in A'
            ∀ x : Sets. x : TA ⇔ x in TA'
            ∀ y : Sets. y : PA ⇔ y in PA'
            ∀ e : Sets. ¬(e in EmpSet)
            ∀ z : Sets. z in PA' ⇔ ∀ p : Sets. (p in z ⇒ p in
                A')
            ∀ x : Sets. x in TA' ⇒ x in PA'
            ∀ r, s : Sets. ∀ q : Sets. (q in r inter s ⇔ q in
                r ∧ q in s)
            ∀ a, b : Sets. (b in Uni a ⇔ exists c : Sets. b in
                c ∧ c in a)
%%  Specific axioms for a A as topological space
            EmptySet in TA'
            A' in TA'
            ∀ a, b : TA. a inter b : TA
            ∀ c : TA. Uni c : TA
%%  Definition of B, TB and PB
```

```
          ∀ b : Sets. b : B ⇔ b in B'
          ∀ x : Sets. x : TB ⇔ x in TB'
          ∀ y : Sets. y : PB ⇔ y in PB'
          ∀ z : Sets. z in PB' ⇔ ∀ b : Sets. (b in z ⇒ b in
            B')
          ∀ k : Sets. k in TB' ⇒ k in PB
%%  Specific axioms for a B as topological space
          EmptySet in TB'
          B' in TB'
          ∀ a, b : TB. a inter b : TB
          ∀ c : TB. Uni c : TB
%%  Inverse image of a set under a function
          ∀ q : TB. ∀ x : Sets.
          x in inversef q ⇔ f(x) in q
%%  Condition of continuity
          ∀ x : PB. (x : TB ⇒ inversef(x) in TA')
end


%%  Perfect square of a topological space

spec PerfSqTopSp =
     sorts  Sets, X, TX, PX, XX, TXX, PXX;
            X, TX, PX, XX, TXX, PXX < Sets;
            TX < PX; TXX < PXX;
     ops    EmpSet, X', TX', PX', XX', TXX', PXX' : Sets;
            __ in __ : Sets × Sets;
            __ subset __ : Sets × Sets;
            __ inter __ : Sets × Sets → Sets
            __ ordpair __ : Sets × Sets → Sets
            __ prod __ : Sets × Sets → Sets
            Uni : Sets → Sets
     preds ∀ x : Sets. x : X ⇔ x in X'
            ∀ u : Sets. u : TX ⇔ u in TX'
            ∀ v : Sets. v : PX ⇔ v in PX'
            ∀ a : Sets. a : XX ⇔ a in XX'
            ∀ b : Sets. b : TXX ⇔ b in TXX'
            ∀ c : Sets. c : PXX ⇔ c in PXX'
            ∀ e : Sets. ¬(e in EmpSet)
```

```
            ∀ z : Sets. z in PX' ⇔ ∀ b : Sets. (b in z ⇒ b in
                X')
            ∀ d : Sets. d in TX' ⇒ d in PX'
            ∀ r, s : Sets. ∀ q: Sets. (q in r inter s ⇔ q in r
                 ∧ q in s)
            ∀ a, b : Sets. (b in Uni a ⇔ ∃ c : Sets. b in c ∧
                c in a)
            ∀ a, b : Sets. a subset b ⇔ (∀ u : Sets. u in a ⇒
                u in b)
%%  Specific axioms for a X as topological space
            EmpSet in TX'
            X' in TX'
            ∀ a, b : TX. a inter b :TX
            ∀ c : TX. Uni c :TX
 %%  Specific axioms for XX as topological space
            EmpSet in TXX'
            XX' in TXX'
            ∀ a, b : TXX. a inter b : TXX
            ∀ c : TXX. Uni c : TXX
%%  Defining ordpair
            ∀ u, x, y : Sets. u in x ordpar y ⇔ ((∀ t: Sets. t
                 in u ⇔ (t = x)) ∨ (∀ s : Sets. s in u ⇔ (s =
                x ∨ s = y)))
%%  Definying prod
            ∀ V, W : Sets.
            ∀ a : Sets. a in V prod W ⇔ ∃ v : V. ∃ w : W. a =
                v ordpair w
%%  Specifying the fact that XX is X prod X
            ∀ x : Sets. x in XX ⇔ (∃ s,t: X. x = s ordpair t)
%%  TXX is the product topology
            ∀ z : Sets. z :TXX ⇔ (∀ w:Sets. w in z. ⇔ (∃ u,v :
                Sets. (u in TX ∧ v in TX ∧ w in u prod v ∧ u
                prod v subset z)))
end

%%  Contiunuous binary operation

spec Generic =
     sorts  GSets, U, UU, TU, TUU, PU, PUU,
```

```
     ops   GEmpt, U', UU', TU', TUU', PU', PUU': GSets
           __ in __ : GSets × GSets;
           __ subset __ : GSets × GSets;
           __ inter __ : GSets × GSets ↦ GSets
           Uni : GSets ↦ GSets
end

view I1:
    Generic to PerfSqTopSp
    GSets ↦ Sets, U ↦ X, UU ↦ XX, TUU ↦ TXX, PU ↦ PX, PUU
        ↦ PXX, U' ↦ X', UU' ↦ XX', TUU' ↦ TXX', PU' ↦ PX',
        PUU' ↦ PXX', GEmpt ↦ EmpSet, __ in __ ↦ __ in __, __
        subset __ ↦ __ subset __, __ inter __ ↦ __ inter __,
        Uni ↦ Uni
end

view I2:
    Generic to ContFunc
    GSets ↦ Sets, U ↦ B, TU ↦ TB, PU ↦ PB, UU ↦ A, TUU ↦
        TA, PUU ↦ PA, U' ↦ B', TU' ↦ TB',
    PU' ↦ PB', UU' ↦ A', TUU' ↦ TA', PUU' ↦ PA', GEmpt ↦
        EmpSet, __ in __ ↦ __ in __, __ subset __ ↦ __ subset
        __, __ inter __ ↦ __ inter __, Uni ↦ Uni
end

spec Colimit = combine I1, I2
```

By computing the corresponding blend (i.e. colimit), we obtain a specification of the notion of continuous binary operation. So, after doing some improvements in the presentation (e.g. updating names of sorts), one essentially obtains an specification like the following:

```
logic CASL

spec ContBinOp =
     sorts  GSets, U, UU, TU, TUU, PU, PUU;
            U, UU, TU, TUU, PU, PUU < GSets;
            TU < PU, TUU < PUU
     ops   GEmpt, U', UU', TU', TUU', PU', PUU': GSets
         __ in __ : GSets × GSets;
```

```
        __ subset __ : GSets × GSets;
        __ inter __ : GSets × GSets ↦ GSets
        Uni : GSets ↦ GSets
        __ ordpair __ : GSets × GSets ↦ GSets
        __ prod __ : GSets × GSets ↦ GSets
        f : UU ↦ U
        inversef : TU ↦ TUU;
    Preds  ∀ x : GSets. x : U ⇔ x in U'
           ∀ w : GSets. w : TU ⇔ w in TU'
           ∀ a : GSets. a : PU ⇔ a in PU'
           ∀ a : GSets. a : PUU ⇔ a in PUU'
           ∀ w : GSets. w : TUU ⇔ w in TUU'
           ∀ y : GSets. ¬(y in EmpSet)
           ∀ z : GSets. z in PU' ⇔ ∀ b : GSets. (b in z ⇒ b
               in U')
           ∀ a : GSets. a in TU' ⇒ a in PU'
           ∀ q, r, s : GSets. (q in r inter s ⇔ q in r ∧ q in
                s)
           ∀ a, b : GSets. (b in Uni a ⇔ ∃ c : GSets. b in c
               ∧ c in a)
           ∀ a, b : GSets. a subset b ⇔ ∀ u : GSets. u in a
               ⇒ u in b)
%%  Specific axioms for U as topological space
           GEmpSet in TU'
           U' in TU'
           ∀ a, b : TU. a inter b : TU
           ∀ c : TU. Uni c : TU
 %%  Specific axioms for UU as topological space
           GEmpSet in TUU'
           UU' in TUU'
           ∀ a, b : TUU. a inter b : TUU
           ∀ c : TUU. Uni c : TUU
%%  Defining ordpair
           ∀ u, x, y : GSets. u in x ordpar y ⇔(∀ t : GSets.
               t in u ⇔ (t = x)) ∨ ∀ s : GSets. s in u ⇔ (s =
                x ∨ s = y)))
%%  Defining prod
           ∀ V, W : GSets.
```

```
            ∀ a : GSets. a in V prod W ⇔ ∃ v : V. ∃ w : W. a =
                 v ordpair w
%%  Specifying the fact that UU is U prod U
            ∀ x : GSets. x in UU ⇔ ∃ s, t : U. x = s ordpair t
%%  TUU is the product topology
            ∀ z : GSets. z : TUU ⇔ ∀ w : GSets. w in z. ⇔ ∃ u,
                 v : GSets. (u in TU ∧ v in TU ∧ w in u prod v
                ∧ u prod v subset z)))
%%  Inverse image of a set under a function
            ∀ q : TUU. ∀ x : GSets. x in inversef(q) ⇔ f(x) in
                 q
%%  Condition of continuity
            ∀ x : PU. x : TU ⇒ inversef(x) in TUU'
end
```

3.2. **Quasi-topological Groups.**

Let us combine the latter blended concept (i.e. continuous binary operations) with (an enriched form of) the notion of group to generate the concept of quasi-topological groups.

```
Login CASL

spec ContBinOp =

...

end

%%  (Enriched) group

spec EnrGroup =
     sorts Sets, L, LS, L < Sets, LS < Sets
     ops  0 : L; L', LS' : Sets
          __ in __ : Sets × Sets;
          __ ordpair __ : Sets × Sets → Sets
          __ prod __ : Sets × Sets → Sets
          __ + __ : L × L → L
          Addinv : L → L
          ++ : LS → L
%%  L' and LS' simulates L and LS
```

```
     preds  ∀ x : Sets. x: L ⇔ x in L'
            ∀ w : Sets. w: LS ⇔ w in LS'
%%  Axioms of a Group.
            ∀ x, y, z : L. (x + y) + z = x + (y + z)
            ∀ x : L. x + 0 = x ∧ 0 + x = x
            ∀ x : L. Addinv(x) + x = 0 ∧ x + Addinv(x) = 0
%%  Defining ordpair
            ∀ u, x, y : Sets. u in x ordpar y ⇔ ((∀ t : Sets.
                t in u ⇔ (t = x)) ∨ (∀ s : Sets. s in u ⇔ (s =
                 x ∨ s = y)))
%%  Defining prod
            ∀ V, W : Sets. ∀ a : Sets. a in V prod W ⇔ ∃ v : V
                . ∃ w: W. a = v ordpair w
%%  Specifying the fact that LS is the cartesian product of L
    with L
            ∀ x, y :Sets. x ordpair y in LS' ⇔ x in L' ∧ y in
                L'
%%  ++ simulates +
            ∀ a, b : Sets. if a : L ∧ b : L. a + b = ++(a
                ordpair b)
end


%%  Quasi-topological group as a blend of ComBinOp and EnrGroup
spec Generic =
     sorts Sets, G, GS
     ops  __ in __ : GS → G
          __ ordpair __ : Sets × Sets → Sets
          __ prod __ : Sets × Sets → Sets
          Addinv : L → L
          ++ : LS → L
end

view I1:
        Generic to EnrGroup
        Sets → Sets, G ↦ L, GS → LS,  ++ → ++
end

view I2:
```

```
        Generic to ContBinOp
Sets ↦ GSets,  __ in __ ↦ __ in __, __ ordpair __  ↦ __
    ordpair __, __ prod __ ↦ __ prod __, ++ __  ↦ f
end

spec Colimit = combine I1, I2
```

After doing the computation of the colimit we obtain the concept of quasi-topological group.

```
spec QuasiTopGr =
     sorts   Sets, G, TG, PG, GS, TGS, PGS;
             G, TG, PG, GS, TGS, PGS < Sets;
             TG < PG, TGS < PGS;
     ops     Empty, G', TG', PG', GS', TGS', PGS';
             __ ordpair __ : Sets × Sets → Sets
             __ prod __ : Sets × Sets → Sets
             __ + __ : G × G → G
             __ in __ : Sets × Sets
             ++ : GS → G
             funcinvplusplus : TG →  TGS
             Addinv : G → G
%%  Axioms for G being a group
    preds ∀ x, y, z : G. (x + y) + z = x + (y + z)
          ∀ x : G. x + 0 = x ∧ 0 + x = x
          ∀ x : G. Addinv(x) + x = 0 ∧ x + Addinv(x) = 0
%%  Axioms for GS being the perfect square topological space
        ∀ g : Sets. g: G  ⇔ g in G'
        ∀ h : Sets. h : TG  ⇔ h in TG'
        ∀ m : Sets. m : PG  ⇔ m in PG'
        ∀ r : Sets. r : GS  ⇔ r in GS'
        ∀ r : Sets. r : TGS  ⇔ r in TGS'
        ∀ n : Sets. n : PGS  ⇔ n in PGS'
        ∀ s : Sets. ¬ (s in Empty)
        ∀ t : Sets. t in PS' ⇔ ∀ b : Sets. (b in t ⇒ b in G')
        ∀ u : Sets. u in TS' ⇒ u in PS'
        ∀ q, u, v : Sets. (q in u inter v ⇔ q in u ∧ q in v)
        ∀ a, b : Sets. (b in Uni a ⇔ ∃ c : Sets. b in c ∧ c in
             a)
        ∀ x, y : Sets. x subset y ⇔ (∀ u : Sets. u in x ⇒ u in
             y)
```

```
%%  Specific axioms for a G as topological space
        Empty in TG'
        G' in TG'
        ∀ a, b : TG. a inter b : TG
        ∀ c : TG. Uni c :TG
 %%  Specific axioms for GS as topological space
        Empty in TGS'
        GS' in TGS'
        ∀ a, b : TGS. a inter b : TGS
        ∀ c: TGS. Uni c :TGS
%%  Defining ordpair
        ∀ u, x, y : Sets. u in x ordpar y ⇔ ((∀ t : Sets. t in
             u ⇔ (t = x)) ∨ (∀ s : Sets. s in u ⇔ (s = x ∨ s
            = y)))
%%  Defining prod
        ∀ V, W : Sets. ∀ k : Sets. k in V prod W ⇔ ∃ v : V. ∃
            w : W. k = v ordpair w
%%  Specifying the fact that GS is G prod G
        ∀ x: Sets. x in GS ⇔ (∃ s,t: G. x = s ordpair t)
%%  TGS is product topology
        ∀ z: Sets. z : TGS ⇔ (∀ w: Sets. w in z. ⇔ (∃ u, v :
            Sets. (u in TS ∧ v in TS ∧ w in u prod v ∧ u prod v
             subset z)))
%%  Inverse image of a set under a function
        ∀ q : TG. ∀ x : Sets. x in invfuncplusplus(q) ⇔  ++(x)
             in q
%%  Condition of continuity
        ∀ x : PGS. (x : TGS ⇒  invfuncplusplus(x) in TG')
```

end

3.3. **Continuous Endomorphisms.**

We will obtain the notion of continuous endmorphism starting with continuous functions (between topological spaces) and doing a conceptual identification between the domain and the codomain of the corresponding map. Explicitly, in the former specification of the conceptual space of continuous functions, we declare the equality of the corresponding sorts of the domain and codomain as follows: $A \cong B; TA \cong TB$ and $PA \cong PB$. In this way, we obtain the concrete specification of the notion of continuous endomorphism:

```
begin CASL

%%  Continuous Endomorphism

spec ContEndo =
     sorts   Sets, A, TA, PA;
             A, TA, PA < Sets;
             TA < PA,
%%  A = domain and codomain of the function, TA = topology of A,
     PA = powerset of A
      ops   EmpSet, A', TA', PA' : Sets;
    __ in __ : Sets × Sets
    __ inter __ : Sets × Sets ↦ Sets
    Uni__ : Sets
    f : A ↦ A
    inversef : TA ↦ TA
%%  Definition of A, TA and PA
      preds ∀ a : Sets. a : A ⇔ a in A'
            ∀ x : Sets. x : TA ⇔ x in TA'
            ∀ y : Sets. y : PA ⇔ y in PA'
            ∀ e : Sets. ¬(e in EmpSet)
            ∀ z : Sets. z in PA' ⇔ ∀ p : Sets. (p in z ⇒ p in
                A')
            ∀ a : Sets. x in TA' ⇒ x in PA'
            ∀ q, r, s : Sets. (q in r inter s ⇔ q in r ∧ q in
                s)
            ∀ a : Sets. ∀ b : Sets. (b in Uni a ⇔ exists c :
                Sets. b in c ∧ c in a)
%%  Specific axioms for a A as topological space
            EmptySet in TA'
            A' in TA'
            ∀ a, b : TA. a inter b : TA
            ∀ c : TA. Uni c : TA
%%  Inverse image of a set under a function
            ∀ q : TA. ∀ x : Sets.
            x in inversef(q) ⇔ f(x) in q
%%  Condition of continuity
            ∀ x : PA. (x : TA ⇒ inversef(x) in TA')
end
```

3.4. Topological Groups.

Finally, we generate the concept of Topological Group as the following blend (i.e. colimit) of the former two (specifications of) concepts; i.e., quasi-topological groups and continuous endomorphisms:

```
Login CASL

spec QuasiTopGr =

...

end


spec ContEndo =

...

end

%%  Topological group as a blend of QuasiTopGr and ContEndo
spec Generic =
     sorts Sets, H, TH, PH, HS, THS, PHS
     ops  Empty, H', TH', PH', HS', THS', PHS': Sets
           __ in __ : Sets × Sets
           __ inter __ :Sets × Sets ↦ Sets
           Uni: Sets ↦ Sets
           Addinv: H → H
end

view I1:
        Generic to QuasiTopGr
        Sets → Sets, H ↦ G, TH → TG, PH ↦ PG, HS ↦ GS, THS
            ↦ TGS, PHS ↦ PGS, Empty ↦ Empty, H' ↦ G', TH'
            → TG', PH' ↦ PG', HS' ↦ GS', THS' ↦ TGS', PHS'
            ↦ PGS', in ↦ in, __ inter __ ↦ __ inter __ , Uni
            ↦ Uni, Addinv ↦ Addinv,
end
```

```
view I2:
        Generic to ContEndo
        Sets ↦ Sets, H ↦ A, TH ↦ TA, PH ↦ PA, Empty ↦
            EmpSet, H' ↦ A', TH' → TA', PH' ↦ PA', in ↦ in,
            __ inter __ ↦ __ inter __, Uni ↦ Uni, Addinv ↦ f
end

spec Colimit = combine I1, I2
```

After doing the computation of the colimit we essentially obtain the classic concept of topological group:

```
spec TopGroup =
     sorts   Sets, H, TH, PH, HS, THS, PHS;
            H, TH, PH, HS, THS, PHS < Sets;
            TH < PH, THS < PHS;
     ops    Empty, H', TH', PH', HS', THS', PHS';
            __ ordpair __ : Sets × Sets → Sets
            __ prod __ : Sets × Sets → Sets
            __ + __ : H × H → H
            __ in __ : Sets × Sets
            ++ : HS → H
            invfuncplusplus : H → HS
            Addinv : H → H
            invfuncAddinv : H → H
%%  Axioms for H being a group
    preds ∀ x, y, z : H. (x + y) + z = x + (y + z)
          ∀ x : H. x + 0 = x ∧ 0 + x = x
          ∀ x : H. Addinv(x) + x = 0 ∧ x + Addinv(x) = 0
%%  Axioms for HS being the  perfect square topological space
        ∀ g : Sets. g : H  ⇔ g in H'
        ∀ h : Sets. h : TH  ⇔ h in TH'
        ∀ m : Sets. m : PH  ⇔ m in PH'
        ∀ r : Sets. r : HS  ⇔ r in HS'
        ∀ r : Sets. r : THS  ⇔ r in THS'
        ∀ n : Sets. n : PHS  ⇔ n in PHS'
        ∀ s : Sets. ¬ (s in Empty)
        ∀ t : Sets. t in PS' ⇔ ∀ b : Sets. (b in t ⇒ b in H')
        ∀ u : Sets. u in TS' ⇒ u in PS'
        ∀ q, u, v : Sets. (q in u inter v ⇔ q in u ∧ q in v)
```

```
        ∀ a, b : Sets. (b in Uni a ⇔ ∃ c : Sets. b in c ∧ c in
             a)
        ∀ x, y : Sets. x subset y ⇔ (∀ u : Sets. u in x ⇒ u in
             y)
%%  Specific axioms for H as topological space
        Empty in TH'
        H' in TH'
        ∀ a, b : TH. a inter b : TH
        ∀ c : TH. Uni c : TH
 %%  Specific axioms for HS as topological space
        Empty in THS'
        HS' in THS'
        ∀ a, b : THS. a inter b : THS
        ∀ c : THS. Uni c : THS
%%  Defining ordpair
        ∀ u, x, y : Sets. u in x ordpar y ⇔ ((∀ t : Sets. t in
             u ⇔ (t = x)) ∨ (∀ s : Sets. s in u ⇔ (s = x ∨ s
            = y)))
%%  Defining prod
        ∀ V, W : Sets. ∀ k : Sets. k in V prod W ⇔ ∃ v : V. ∃
            w : W. k = v ordpair w
%%  Specifying the fact that HS is H prod H
        ∀ x : Sets. x in HS ⇔ (∃ s, t : H. x = s ordpair t)
%%  THS is product topology
        ∀ z : Sets. z : THS ⇔ (∀ w : Sets. w in z. ⇔ (∃ u, v :
             Sets. (u in TS ∧ v in TS ∧ w in u prod v ∧ u prod
            v subset z)))
%%  Inverse image of a set under a function
        ∀ q : TH. ∀ x : Sets. x in invfuncplusplus q ⇔
            invfuncplusplus(x) in q
%%  Condition of continuity
        ∀ x : PHS. (x : THS ⇒  invfuncplusplus(x) in TH')
%%  Inverse image of a set under a Addinv
          ∀ q : TH. ∀ x : Sets.
            x in invfuncAddinv(q) ⇔ Addinv(x) in q
%%condition of continuity
            ∀ x : PH. (x : TH ⇒ invfuncAddinv(x) in TH')
```

end

The former specification is considerably larger than the one usually given in the text books due to the fact that we include additionally the minimal set-theoretical information required to define an essentially autonomous concept which can be coherently described with the semantic tools of CASL and HETS. In fact, the former conceptual computation were explicitly run and proved in HETS.

We also can state the results presented in the former specifications in the form of a global (meta-)theorem describing categorical constructions done before in terms of colimits:

Theorem 3.1. *The concept of topological group, viewed as a unique object of the co-complete category of many-sorted first-order theories with axiom-preserving signature morphisms, can be generated recursively by means of three formal colimits (blends), starting from the concepts of (enriched groups, continuous functions and continuous endomorphisms).*

4. Conclusions

In Figure 4, we present a diagrammatic summary of the whole recursive generation done through formal conceptual blending with the help of conceptual identification.

The fact that we explicitly find artificial specifications of sophisticated concepts in abstract algebra and topology represents valuable domain-specific evidence for the universality of the meta-tools described by means of categorical formalizations of conceptual blending (and, in an indirect way, by the more informal categorical approach of conceptual identification made in terms of sorts' identifications).

The former results also promote the thesis that the potential scope of the co-creative power of artificial interactive systems regarding mathematical invention goes beyond the typical elementary structures classically studied, e.g., the complex numbers [8].

Finally, this research goes towards the development of new forms of conceptual co-creative cybernetics in the domain of interactive mathematical creation. In fact, previous forms of this new kind of cybernetics were developed within the multidisciplinary research consortium COINVENT [17], [6]. Specifically, in [5], an interactive co-creative computational prototype called COBBLE is presented, materializing artificial co-innovative reasoning in mathematics and music (harmonization) based on notions coming not only from conceptual blending theory, but also from analogical reasoning, formal ontology theory, logic programming and formal methods. So, the collection of results presented here can be seen as a first theoretical step towards extensions of such computational prototypes to broader mathematically-based disciplines.

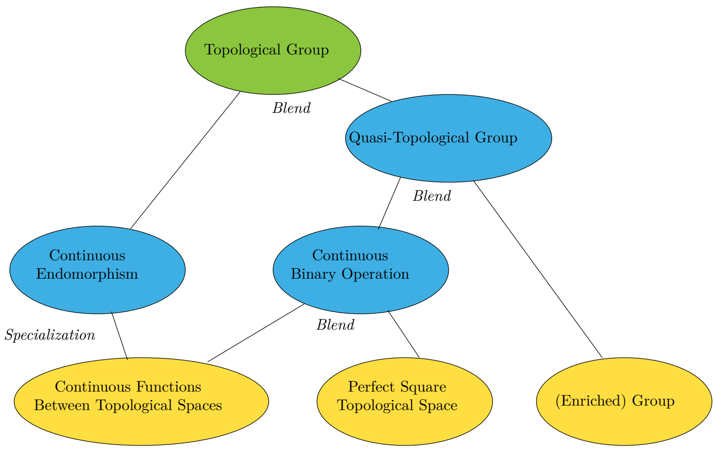

Figure 1. Diagrammatic Representation for the recursive generation of the concept of Topological Group through Formal Conceptual Blending and specialization

Acknowledgements

The first two authors thank the Universidad Autonoma de Bucaramanga for all the support (Grant I56073).

References

[1] James C Alexander. "Blending in mathematics". In: *Semiotica* 2011.187 (2011), pp. 1–48.

[2] Michel Bidoit and Peter D. Mosses. *CASL User Manual*. Lecture Note in Computer Science 2900, Springer-Verlag, Berlin, Heidelberg, 2004.

[3] F. Bou et al. "The role of blending in mathematical invention". In: *Proceedings of the Sixth International Conference on Computational Creativity (ICCC). S. Colton et. al., eds. Park City, Utah, June 29-July 2, 2015. Publisher: Brigham Young University, Provo, Utah.* (2015), pp. 55–62.

[4] M. Codescu et al. "Conceptual Blending in DOL: Evaluating Consistency and Conflict resolution". In: *Concept Invention: Foundations, Implementation, Social Aspects and Applications*. Ed. by R. Confalonieri et al. Cham: Springer International Publishing, 2018, pp. 69–96.

[5] R. Codescu. M. Confalonieri and M. Eppe. "Enabling Technologies for Conceptual Invention". In: *Concept Invention: Foundations, Implementation, Social Aspects and Applications*. Ed. by R. Confalonieri et al. Cham: Springer International Publishing, 2018, pp. 189–219.

[6] In: *Concept Invention: Foundations, Implementation, Social Aspects and Applications*. Ed. by R. Confalonieri et al. Cham: Springer International Publishing, 2018.
[7] G. Fauconnier and M. Turner. *The Way We Think*. Basic Books, 2003.
[8] J. Fleuriot et al. "Reinventing the complex numbers". In: vol. 1. PICS. Workshop at ECAE, Prague. Workshop on Computational Creativity, Concept Invention, and General Intelligence. Tarek B. Besold et al. (Eds) Publications of the Institute of Cognitive Sciences, PICS series, University of Osnabrück. 2014.
[9] John B Fraleigh. *A first course in abstract algebra*. Pearson Education India, 2003.
[10] D.A.J. Gomez-Ramirez and Alan Smaill. "Formal Conceptual Blending in the (Co-)Invention of (Pure) Mathematics". In: *Concept Invention: Foundations, Implementation, Social Aspects and Applications*. Ed. by R. Confalonieri et al. Cham: Springer International Publishing, 2018, pp. 221–239.
[11] D.A.J. Gomez-Ramirez et al. "Cagegory-Based Co-Generation of Seminal Concepts and Results in Algebra and Number Theory: Containment Division and Goldbach Rings". In: *JP Journal of Algebra, Number Theory and Applications* 40.5 (2018), pp. 887–901.
[12] Robert C Holte. "A conceptual framework for concept identification". In: *Machine Learning*. Springer, 1986, pp. 99–102.
[13] I Kurcz, GW Shugar, and JH Danks. "A simulation approach to conceptual identification". In: *Knowledge and Language* 39 (1986), p. 49.
[14] M. Martinez et al. "Theory Blending: Extended Algorithmic Aspects and Examples". In: *Annals of Mathematics and Artificial Intelligence* (2016), pp. 1–25.
[15] T. Mossakowski, C. Maeder, and M. Codescu. "HETS User Guide (Version 0,99)". In: (2014). URL: `www.informatik.uni-bremen.de/agbkb/forschung/formal%5C_methods/CoFI/hets/UserGuide.pdf`.
[16] James R Munkres. *Topology*. Prentice Hall, 2000.
[17] M. Schorlemmer et al. "COINVENT: Towards a Computational Concept Invention Theory". In: *5th International Conference on Computational Creativity (ICCC)*.
[18] A. Schwering et al. "Syntactic Principles of Heuristic Driven Theory Projection". In: *Cognitive Systems Research* 10(3) (2009), pp. 251–269.

Vienna University of Technology, Institute of Discrete Mathematics and Geometry, wiedner Hauptstasse 8-10, 1040, Vienna, Austria.

Departamento de Matemáticas y Ciencias Naturales, Universidad Autónoma de Bucaramanga, Bucaramanga, Santander 680003, Colombia.

University of Antioquia, Calle 67 # 53-108, Medellín, Colombia

Institute of Discrete Mathematics and Geometry, Vienna University of Technology Vienna, Austria.
E-mail address: daj.gomezramirez@gmail.com
E-mail address: yherrera743@unab.edu.co
E-mail address: fgeismann@live.at