

Formal Adventures in Convex and Conical Spaces

Reynald Affeldt^{[0000-0002-2327-953X]1}, Jacques Garrigue^{[0000-0001-8056-5519]2},
and Takafumi Saikawa^{[0000-0003-4492-745X]2}

¹ National Institute of Advanced Industrial Science and Technology, Japan

² Nagoya University, Japan

Abstract. Convex sets appear in various mathematical theories, and are used to define notions such as convex functions and hulls. As an abstraction from the usual definition of convex sets in vector spaces, we formalize in Coq an intrinsic axiomatization of convex sets, namely convex spaces, based on an operation taking barycenters of points. A convex space corresponds to a specific type that does not refer to a surrounding vector space. This simplifies the definitions of functions on it. We show applications including the convexity of information-theoretic functions defined over types of distributions. We also show how convex spaces are embedded in conical spaces, which are abstract real cones, and use the embedding as an effective device to ease calculations.

1 Introduction

The notion of convex sets appears in various mathematical theories. A subset X of a real vector space is called a convex set if, for any $x, y \in X$ and $p \in [0, 1]$, their *convex combination* $px + (1 - p)y$ is again in X . One basic use of it is to define the convexity of functions. A function f is said to be convex if $f(px + (1 - p)y) \leq pf(x) + (1 - p)f(y)$ for any convex combination $px + (1 - p)y$. Thus, convex sets are natural domains for convex functions to be defined on. Good examples of these notions can be found in information theory, where convexity is a fundamental property of important functions such as logarithm, entropy, and mutual information. Our INFO_{THEO} library [17] developed in the COQ proof assistant [29] has a formalization of textbook proofs [12] of such results.

In the course of formalizing such convexity results, we find that axiomatizing convex sets is a useful step which provides clarity and organizability in the results. We abstract the usual treatment of convex sets as subsets of some vector space and employ an algebraic theory of *convex spaces*, which was introduced by Stone [27]. The formalization uses the *packed class* construction [15, 24], so as to obtain generic notations and lemmas, and more importantly, to be able to combine structures. Binary convex spaces are formalized in Sect. 2, and their multiary versions are formalized in Sect. 3, along with proofs of equivalence.

We also formalize an embedding of convex spaces into *conical spaces* (a.k.a. cones or real cones [31]), which we find an indispensable tool to formalize convex

spaces. Examples in the literature avoid proving properties of convex spaces directly and choose to work in conical spaces. This is especially the case when their goal can be achieved either way [23, 31]. Some authors suggest that the results in conical spaces can be backported to convex spaces [13, 21]. We apply this method in Sect. 4 to enable additive handling of convex combinations. By formalizing the relationship between convex and conical spaces, we work out short proofs of a number of lemmas on convex spaces. Among them is Stone’s key lemma [27, Lemma 2], whose proof is often omitted in the literature despite its fundamental role in the study of convex spaces.

We complete this presentation with applications of our formalization to convex hulls (Sect. 5) and to convex functions (Sect. 6).

While our proofs do not introduce extra axioms, some libraries used in our development, such as `mathcomp-analysis` [1], contain axioms which make parts of our work classical. In particular, our definition of convex sets is based on classical sets, assuming decidable membership.

2 Convex spaces

Let us begin with the definition of convex spaces. As mentioned in the introduction, convex spaces are an axiomatization of the usual notion of convex sets in vector spaces. It has a long history of repeated reintroduction by many authors, often with minor differences and different names: barycentric algebra [27], semiconvex algebra [28], or, just, convex sets [19].

We define convex spaces following Fritz [14, Definition 3.1].

Definition 1 (Module `ConvexSpace` in [18]). *A convex space is a structure for the following signature:*

- Carrier set X .
- Convex combination operations $(_ \triangleleft_p \triangleright _) : X \times X \rightarrow X$ indexed by $p \in [0, 1]$.
- Unit law: $x \triangleleft_1 \triangleright y = x$.
- Idempotence law: $x \triangleleft_p \triangleright x = x$.
- Skewed commutativity law: $x \triangleleft_{1-p} \triangleright y = y \triangleleft_p \triangleright x$.
- Quasi-associativity law: $x \triangleleft_p \triangleright (y \triangleleft_q \triangleright z) = (x \triangleleft_r \triangleright y) \triangleleft_s \triangleright z$,

$$\text{where } s = 1 - (1 - p)(1 - q) \text{ and } r = \begin{cases} p/s & \text{if } s \neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

(Note that r is irrelevant to the value of $(x \triangleleft_r \triangleright y) \triangleleft_s \triangleright z$ if $s = 0$.)

We can translate this definition to COQ as a packed class [15] with the following mixin interface:

```

1 Record mixin_of (T : choiceType) : Type := Mixin {
2   conv : prob -> T -> T -> T where "a <| p |> b" := (conv p a b);
3   _ : forall a b, a <| 1%:pr |> b = a ;
4   _ : forall p a, a <| p |> a = a ;
5   _ : forall p a b, a <| p |> b = b <| p.~%:pr |> a;
6   _ : forall (p q : prob) (a b c : T),
7     a <| p |> (b <| q |> c) = (a <| [r_of p, q] |> b) <| [s_of p, q] |> c }.

```

There are some notations and definitions to be explained. The type `prob` in the above COQ code denotes the closed unit interval $[0, 1]$. The notation `r%:pr` is a notation for a real number `r` equipped with a canonical proof that $0 \leq r \leq 1$. The notation `p.~` is for $1 - p$. The notation `[s_of p, q]` is for $1 - (1 - p)(1 - q)$, and `[r_of p, q]` for $p/[s_of p, q]$.

Intuitively, one can regard the convex combination as a probabilistic choice between two points. At line 3, the left argument is chosen with probability 1. The lines that follow correspond to idempotence, skewed commutativity, and quasi-associativity.

An easy example of convex space is the real line \mathbb{R} , whose convex combination is expressed by ordinary addition and multiplication as $pa + (1 - p)b$. Probability distributions also form a convex space. In the formalization, the type `fdist A` of distributions over any finite type `A` (borrowed from previous work [6]) is equipped with a convex space structure, where the convex combination of two distributions d_1, d_2 is defined pointwise as $x \mapsto pd_1(x) + (1 - p)d_2(x)$.

As a result of the packed class construction, we obtain the type `convType` of all types which implicitly carry the above axioms. Then, each example of convex space is declared to be canonically a member of `convType`, enabling the implicit inference of the appropriate convex space structure. These two implicit inference mechanisms combined make the statement of generic lemmas on convex spaces simple and applications easy.

3 Multiary convex combination

Convex spaces can also be characterized by multiary convex combination operations, which combine finitely many points x_0, \dots, x_{n-1} at once, according to some finite probability distribution d over the set $I_n = \{0, \dots, n - 1\}$, i.e., $d_i \geq 0$ and $\sum_{i < n} d_i = 1$. In this section we consider different axiomatizations, and their equivalence with the binary axioms.

3.1 Axiomatization

A definition of convex spaces based on multiary operations is given as follows (see for example [10, Definition 5] and [16, Sect. 2.1]).

Definition 2 (Convex space, multiary version). *A convex space based on multiary operations is a structure for the following signature:*

- Carrier set X .
- Multiary convex combination operations, indexed by an arity n and a distribution d over I_n :

$$\begin{aligned} X^n &\rightarrow X \\ (x_i)_{i < n} &\mapsto \langle\!\langle\! d_i x_i \end{aligned}$$

- Projection law: if $d_j = 1$, $\langle\!\langle\! d_i x_i = x_j$. (ax_proj in [18])

$$- \text{Barycenter law: } \langle\!\langle\!_{i<n} d_i \left(\langle\!\langle\!_{j<m} e_{i,j} x_j \right) = \langle\!\langle\!_{j<m} \left(\sum_{i<n} d_i e_{i,j} \right) x_j. \text{ (ax_bary in [18])}$$

Note that in our COQ code, $\langle\!\langle\!_{i<n} d_i x_i$ appears as `<&>_d x` or `altConv d x`, indicating more explicitly that the operation takes two arguments d and x .

This multiary convex structure and the binary one given in Sect. 2 are equivalent: the multiary and binary operators interpret each other satisfying the needed axioms, and the interpretations cancel out when composed. While the binary axiomatization is easy to instantiate, the multiary version exhibits the relationship to probability distributions. Therefore we want to establish this equivalence before working further on other constructions over convex spaces.

In the literature, this equivalence is justified without much detail by referring to the seminal article by Stone [27] (see, e.g., [19, Theorem 4], [10, Proposition 7]). Yet, what Stone gave is not an explicit axiomatization of the multiary convex operator, but a number of lemmas targeted at proving an embedding of (binary) convex spaces into vector spaces. These lemmas include the following one, that is seen as a justification for the barycenter law in the binary axiomatization.

Lemma 1 (Lemma 4 in [27]). *If the given masses and their associated points are partitioned into groups (of non-zero total masses) in any way, then the center of mass is identical with that of masses equal to the respective total masses for the various groups, each placed at the center of mass for the corresponding group.*

The relation to the barycenter law is implied if one sees a convex combination $\langle\!\langle\!_{j<m} (\sum_{i<n} d_i e_{i,j}) x_j$ as a point defined in terms of a set of generating points $\{x_j\}_{j<m}$ (they generate their convex hull). Then $\langle\!\langle\!_{i<n} d_i (\langle\!\langle\!_{j<m} e_{i,j} x_j)$ corresponds to grouping the generating points by filtering through the distributions $\{e_i\}_{i<n}$. But this grouping is not necessarily a partition since there could be shared elements, hence the relation is not direct.

Beaulieu [8, Def. 3.1.4] proposed an alternative multiary axiomatization, which was actually presented as a model for countable probabilistic choice (rather than a definition of convex space). His partition law corresponds exactly to the statement of Stone's lemma.

Definition 3 (Convex space, Beaulieu style). *A convex space is a structure for the previous operations $\langle\!\langle\!_{i<n} d_i$ and the following laws.*

$$- \text{Partition law: } \langle\!\langle\!_{i \in I} \lambda_i x_i = \langle\!\langle\!_{j \in J} \rho_j \left(\langle\!\langle\!_{k \in K_j} \frac{\lambda_k}{\rho_j} x_k \right) \quad (\text{ax_part in [18]})$$

where $\{K_j \mid j \in J\}$ is a partition of I , and $\rho_j = \sum_{k \in K_j} \lambda_k \neq 0$.

$$- \text{Idempotence law: } \langle\!\langle\!_{i \in I} \lambda_i A_i = A \quad \text{if } A_i = A \text{ for all } \lambda_i > 0. \text{ (ax_idem in [18])}$$

In the implementation, using sets as indexing domains of the combination operators would be cumbersome, so that the partition law is actually expressed as follows, using a map \check{K} and Kronecker's δ .

$$\langle\!\langle\!_{i<n} \lambda_i x_i = \langle\!\langle\!_{j<m} \rho_j \left(\langle\!\langle\!_{k<n} \delta_{j, \check{K}(k)} \frac{\lambda_k}{\rho_j} x_k \right) \quad \text{where } \check{K} : I_n \rightarrow I_m, K_j = \check{K}^{-1}(j)$$

We also have to separately show that $(\delta_{j, \tilde{K}(k)} \frac{\lambda_k}{\rho_j})_{k < n}$ and $(\rho_j)_{j < m}$ form probability distributions. As an exceptional case, $(\delta_{j, \tilde{K}(k)} \frac{\lambda_k}{\rho_j})_{k < n}$ is replaced by a uniform distribution if $\rho_j = 0$.

3.2 Equivalence of axiomatizations

After considering the different axiomatizations, we decided to prove a triangular equivalence: between multiary convex structures in standard and Beaulieu style, and then with the binary convex structure given in Sect. 2. The relations we will explain in this section are depicted in Fig. 1.

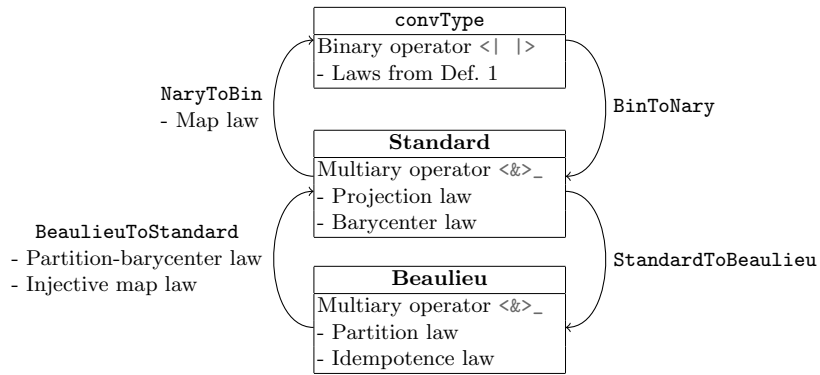


Fig. 1. Relations between the various formalizations of convex spaces

The first equivalence, between multiary convex axioms, is rather technical. The first direction, proving Beaulieu’s axioms from the standard presentation (functor `StandardToBeaulieu` in [18]), is relatively easy, as the partition law is intuitively just a special case of the barycenter law, where supports³ are disjoint, and the idempotence law can be derived as a combination of the two standard laws. However, the second direction (functor `BeaulieuToStandard`) is harder, and led us to introduce two derived laws:

- *Partition-barycenter law: barycenter law, with disjoint supports. (`ax_bary_part`)*
- *Injective map law: $\bigtriangleleft_{i < m} d_i g_{u(i)} = \bigtriangleleft_{j < n} \sum_{\substack{i < m \\ u(i)=j}} d_i g_j$ with u injective. (`ax_inj_map`)*

The partition-barycenter law can be derived from the Beaulieu style axioms, and in turn is used to prove the injective map law. Together they allow to prove the barycenter law.

The equivalence between binary and multiary axiomatizations requires first to define their operators in terms of each other.

³ The support of a probability distribution d is the set $\{i \mid d_i > 0\}$.

Definition 4 (Conv and binconv in [18]).

- (a) Let $d : I_n \rightarrow [0, 1]$ be a finite distribution, and $x : I_n \rightarrow X$ be points in a convex space X . Then the multiary convex combination of these points and distribution is defined from the binary operator by recursion on n as follows:

$$\langle \! \langle \! \langle_{i < n} d_i x_i = \begin{cases} x_0 & \text{if } d_0 = 1 \text{ or } n = 1 \\ x_0 \triangleleft_{d_0} \triangleright \left(\langle \! \langle_{i < n-1} d'_i x_{i+1} \right) & \text{otherwise} \end{cases}$$

where d' is a new distribution: $d'_i = d_{i+1}/(1 - d_0)$.

- (b) Let p be a probability and x_0, x_1 be points in a convex space. Then their binary combination is defined from the multiary operator as follows:

$$x_0 \triangleleft_p \triangleright x_1 = \langle \! \langle_{i < 2} d_i x_i \quad \text{where } d_0 = p \text{ and } d_1 = 1 - p.$$

The first direction, functor **BinToNary** in [18], must prove that the first definition satisfies the multiary axioms, and indeed amounts to proving a variant of Stone's lemma. We will see in the next section that the original proof by Stone is better formalized by transporting the argument to conical spaces.

The opposite direction, functor **NaryToBin**, must prove the binary axioms from the multiary ones. While we start from the standard version, the idempotence law proved to be instrumental in this task, together with the following unrestricted map law.

$$- \text{Map law: } \langle \! \langle_{i < m} d_i g_{u(i)} = \langle \! \langle_{j < n} \sum_{\substack{i < m \\ u(i)=j}} d_i g_j \text{ for any map } u. \quad (\text{ax_map in [18]})$$

Finally, one also needs to prove that the definitions we used for each operation in both directions are coherent.

Lemma 2 (equiv_conv and equiv_convn in [18]). *The constructions in Def. 4 (Conv and binconv) cancel each other. That is,*

- If $\langle \! \langle^*$ is the operator induced by Def. 4(a), and $\triangleleft_{\triangleright}^\dagger$ the one induced from it by Def. 4(b), we can derive $a \triangleleft_p \triangleright^\dagger b = a \triangleleft_p \triangleright b$ from the binary axioms.
- If $\triangleleft_{\triangleright}^*$ the operator induced by Def. 4(b), $\langle \! \langle^\dagger$ is the one induced from it by Def. 4(a), we can derive $\langle \! \langle_{i < n}^\dagger d_i x_i = \langle \! \langle_{i < n} d_i x_i$ from the multiary axioms.

4 Conical spaces and embedded convex spaces

The definition of multiary convex combination operator in the previous section (Def. 4(a)) relied on recursion. This makes the definition look complicated, and moreover, the algebraic properties of the combination difficult to see. If we consider the special case of convex sets in a vector space, the meaning of multiary combinations and the algebraic properties become evident:

$$\langle \! \langle_{i < n} d_i x_i = d_0 x_0 + \cdots + d_{n-1} x_{n-1}.$$

The additions on the right-hand side are of vectors, and thus are associative and commutative. This means that the multiary combination on the left-hand side is invariant under permutations or partitions on indices. We want to show that these invariance properties are also satisfied generally in any convex space.

However, the search for the proofs is painful if naively done. This is because binary convex combination operations satisfy associativity and commutativity only through cumbersome parameter computations. For example, a direct proof of the permutation case involves manipulations on the set I_n of indices and on the symmetry groups, which require fairly long combinatorics [27, Lemma 2].

We present a solution to this complexity by transporting the arguments on convex spaces to a closely related construction of conical spaces. Conical spaces are an abstraction of cones in real vector spaces just like convex spaces are an abstraction of convex sets. Like convex spaces, the definition of conical spaces appears in many articles. We refer to the ones by Flood (called semicone there) [13] and by Varacca and Winskel (called real cone there) [31]:

Definition 5 (Conical space). *A conical space is a semimodule over the semiring of non-negative reals. That is, it is a structure for the following signature:*

- Carrier set X .
- Zero $\mathbf{0} : X$.
- Addition operation $_ + _ : X \times X \rightarrow X$.
- Scaling operations $c \cdot _ : X \rightarrow X$ indexed by $c \in \mathbb{R}_{\geq 0}$.
- Associativity law for addition: $x + (y + z) = (x + y) + z$.
- Commutativity law for addition: $x + y = y + x$.
- Associativity law for scaling: $c(dx) = (cd)x$.
- Left-distributivity law: $(c + d)x = cx + dx$.
- Right-distributivity law: $c(x + y) = cx + cy$.
- Zero law for addition: $\mathbf{0} + x = x$.
- Left zero law for scaling: $0x = \mathbf{0}$.
- Right zero law for scaling: $c\mathbf{0} = \mathbf{0}$.
- One law for scaling: $1x = x$.

We display this definition only to show that conical spaces have straightforward associativity and commutativity. In fact, the formalization is elaborated on the embedding of convex spaces into canonically constructed conical spaces, which appeared in the article by Flood [13]. We build on top of each convex space X , the conical space S_X of its “scaled points”:

Definition 6 (scaled_pt, addpt, and scalept in [18]). *Let X be a convex space. We define a set S_X which becomes a conical space with the following addition and scaling operations.*

$$S_X := (\mathbb{R}_{>0} \times X) \cup \{\mathbf{0}\}.$$

*That is, the points of S_X are either a pair $p * x$ of $p \in \mathbb{R}_{>0}$ and $x \in X$, or a new additive unit $\mathbf{0}$. Addition of points $a, b \in S_X$ is defined by cases to deal with $\mathbf{0}$:*

$$a + b := \begin{cases} (r + q) * (x \triangleleft_{r/(r+q)} \triangleright y) & \text{if } a = r * x \text{ and } b = q * y \\ a & \text{if } b = \mathbf{0} \\ b & \text{if } a = \mathbf{0} \end{cases}$$

Scaling $a \in S_X$ by $p \in \mathbb{R}_{\geq 0}$ is also defined by cases:

$$pa := \begin{cases} pq * x & \text{if } p > 0 \text{ and } a = q * x \\ \mathbf{0} & \text{otherwise} \end{cases}$$

We omit here the proofs that S_X with these addition and scaling satisfies the conical laws. They are proved formally in [18] (see the lemmas `addptC`, `addptA`, `scalept_addpt`, etc.).

Properties of the underlying convex spaces are transported into and back from this conical space, through an embedding:

Definition 7 (`S1` in [18]).

$$\begin{aligned} \iota : X &\mapsto S_X \\ x &\mapsto 1 * x \end{aligned}$$

Convex combinations in X are mapped by ι to additions in S_X .

Lemma 3 (`S1_convn` in [18]).

$$\iota(\langle \! \langle \! \rangle \! \rangle_{i < n} d_i x_i) = \sum_{i < n} d_i \iota(x_i).$$

The right-hand side of the lemma is a conical sum (Fig. 2), which behaves like an ordinary linear sum thanks to the conical laws, and enjoys good support from MATHCOMP's big operator library [9].

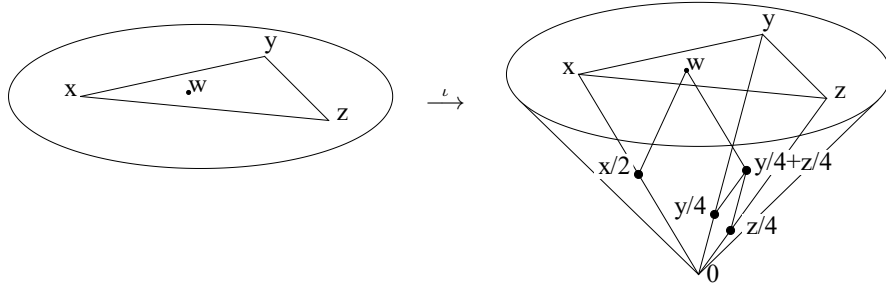


Fig. 2. Example of `S1_convn`: $1 * w = \frac{1}{2} * x + \frac{1}{4} * y + \frac{1}{4} * z$

With these preparations, properties such as [27, Lemma 2] can be proved in a few lines of COQ code:

Lemma 4 (`Conv_n_perm` in [18]).

$$\langle \! \langle \! \rangle \! \rangle_{i < n} d_i x_i = \langle \! \langle \! \rangle \! \rangle_{i < n} (d \circ s)_i (x \circ s)_i,$$

where s is any permutation on the set of indices n .

The proof of the barycenter property [27, Lemma 4] from Sect. 3 is based on the same technique (see `Conv_n_convnfdist` in [18]).

A way to understand this conical approach is to start from Stone’s definition of convex spaces [27]. He uses a quaternary convex operator $(x, y; \alpha, \beta)$ where x and y are points of the space, and α and β are non-negative coefficients such that $\alpha + \beta > 0$. Its values are quotiented by an axiom to be invariant under scaling, removing the need to normalize coefficients for associativity. This amounts to regarding a convex space as the projective space of some conical space.

The definition of S_X is a concrete reconstruction of such a conical space from a given convex space X . The benefit of this method over Stone’s is the removal of quotients by moving the coefficients from operations to values. We can then use the linear-algebraic properties of conical sums such as the neutrality of zeroes, which had to be specially handled in Stone’s proofs (e.g., [27, Lemma 2]).

Examples We illustrate how ι is used in practice with the proof of the *entropic identity*. Let \mathbf{T} be a `convType`; we want to show that

$$(a \triangleleft_q \triangleright b) \triangleleft_p \triangleright (c \triangleleft_q \triangleright d) = (a \triangleleft_p \triangleright c) \triangleleft_q \triangleright (b \triangleleft_p \triangleright d). \quad (1)$$

We could use the properties of convex spaces, but this will result in cumbersome computations, in particular because of quasi-associativity. Instead, we proceed by an embedding into the set of scaled points over \mathbf{T} using ι . First, we observe that these scaled points form a convex space for the operator $p, a, b \mapsto pa + \bar{p}b$ and that $\iota(a \triangleleft_p \triangleright b) = \iota(a) \triangleleft_p \triangleright \iota(b)$. As a consequence, when we apply ι to Equation (1), its left-hand side becomes

$$p(q\iota(a) + \bar{q}\iota(b)) + \bar{p}(q\iota(c) + \bar{q}\iota(d)).$$

The main difference with Equation (1) is that $+$ (COQ notation: `addpt`) enjoys (unconditional) associativity, making the rest of the proof easier. In the proof script below, line 4 performs the embedding by first using the injectivity of ι (lemma `S1_inj`), then using the fact that ι is a morphism w.r.t. $_{-}\triangleleft_p\triangleright_{-}$ (lemma `S1_conv`), and last by revealing the definition of the operator of the convex spaces formed by scaled points (lemma `convptE`). The proof can be completed by rewritings with properties of `addpt` and `scalept` until the left-hand side matches the right-hand side.

```

1 Lemma convACA (a b c d : T) p q :
2   (a <|q|> b) <|p|> (c <|q|> d) = (a <|p|> c) <|q|> (b <|p|> d).
3 Proof.
4 apply S1_inj; rewrite ![in LHS]S1_conv !convptE.
5 rewrite !scalept_addpt ?scalept_comp //.
6 rewrite !(mulRC p) !(mulRC p.~) addptA addptC (addptC (scalept (q * p) _)).
7 rewrite !addptA -addptA -!scalept_comp -?scalept_addpt //.
8 by rewrite !(addptC (scalept _ .~ _)) !S1_conv.
9 Qed.
```

We conclude this section with an example that provides a closed formula for the multiary convex combination $\triangleleft_{i < n} e_i g_i$ (COQ notation: `<|>_e g`) in the case of the real line (seen as a convex space):

```

1 Definition scaleR x : R := if x is p *: y then p * y else 0.
2 Definition big_scaleR := big_morph scaleR scaleR_addpt scaleR0.
3 Lemma avgnRE n (g : 'I_n -> R) e : <|>_e g = \sum_(i < n) e i * g i.
4 Proof.
5 rewrite -[LHS]Scaled1RK S1_convn big_scaleR.
6 by under eq_bigr do rewrite scaleR_scalept // Scaled1RK.
7 Qed.

```

This corresponds to the following transformations of the left-hand side.

$$\begin{aligned}
\langle \rangle_{i < n} e_i g_i &= \text{scaleR}(\iota(\langle \rangle_{i < n} e_i g_i)) && \text{by Scaled1RK} \\
&= \text{scaleR}(\sum_{i < n} e_i \iota(g_i)) && \text{by S1_convn} \\
&= \sum_{i < n} \text{scaleR}(e_i \iota(g_i)) && \text{by big_scaleR} \\
&= \sum_{i < n} e_i \text{scaleR}(\iota(g_i)) && \text{by scaleR_scalept} \\
&= \sum_{i < n} e_i g_i && \text{by Scaled1RK}
\end{aligned}$$

5 Formalization of convex sets and hulls

Our first application of convex and conical spaces is the formalization of convex sets and convex hulls. Besides mathematics, they also appear in many applications of convex spaces such as program semantics [8, 11].

Definition 8 (`is_convex_set` in [18]). *Let T be a convex space. A subset D in T is a convex set if, for any $p \in [0, 1]$ and $x, y \in D$, $x \triangleleft_p \triangleright y \in D$.*

We use the predicate `is_convex_set` to define the type `{convex_set T}` of convex sets over T .

We can turn any set of points in a convex space into a convex set, namely, by taking *convex hulls*.

Definition 9 (`hull` in [18]). *For a subset X of T , its hull \bar{X} is*

$$\bar{X} = \left\{ \langle \rangle_{i < n} d_i x_i \mid n \in \mathbb{N} \wedge d \text{ is a distribution over } I_n \wedge \forall i < n, x_i \in X \right\}.$$

Example The following example illustrates the usefulness of conical spaces when reasoning about convex hulls.

Our goal is to prove that for any $z \in \text{hull}(X \cup Y)$ ($X \neq \emptyset, Y \neq \emptyset$), there exist $x \in X$ and $y \in Y$ such that $z = x \triangleleft_p \triangleright y$ for some p (see the formal statement at line 1 below).

We first introduce two notations. Let `scaled_set X` be the set $\{p * x \mid x \in X\}$. For any $a \neq 0$, let `[point of a0]` (where `a0` is the proof that $a \neq 0$) be the x such that $a = p * x$ for some p .

To prove our goal, it is sufficient to prove that there exist $a \in \text{scaled_set } X$ and $b \in \text{scaled_set } Y$ such that $\iota(z) = a + b$ (this reasoning step is the purpose of line 6). When $a = 0$ or $b = 0$, we omit easy proofs at lines 8 and 9. Otherwise, we can take x to be `[point of a0]` and y to be `[point of b0]` as performed by the four lines from line 10.

We now establish the sufficient condition (from line 14). Since z is in the hull, we have a distribution d and n points g_i such that $z = \langle \rangle_{i < n} d_i g_i$. We then decompose $\iota(z)$ as follows:

$$\iota(z) = \sum_{i < n} d_i(\iota(g_i)) = \underbrace{\sum_{i < n, g_i \in X} d_i(\iota(g_i))}_b + \underbrace{\sum_{i < n, g_i \notin X} d_i(\iota(g_i))}_c.$$

We conclude by observing that b is in `scaled_set X` and that c is in `scaled_set Y` because $\{g_i | g_i \notin X\} \subseteq Y$.

```

1 Lemma hull_setU (z : T) (X Y : {convex_set T}) : X !=set0 -> Y !=set0 ->
2   hull (X `|` Y) z ->
3     exists2 x, x \in X & exists2 y, y \in Y & exists p, z = x <| p |> y.
4 Proof.
5 move=> [dx ?] [dy ?] [n -[g [d [gT zg]]]].
6 suff [a] : exists2 a, a \in scaled_set X & exists2 b, b \in scaled_set Y &
7   S1 z = addpt a b.
8   have [/eqP -> _ [b bY] | a0 aX [b]] := boolP (a == Zero) by ...
9   have [/eqP -> _ | b0 bY] := boolP (b == Zero) by ...
10  rewrite addptE => -[_ zxy].
11  exists [point of a0]; first exact: (@scaled_set_extract _ a).
12  exists [point of b0]; first exact: scaled_set_extract.
13  by eexists; rewrite zxy.
14 move/(congr1 (@S1 _)): zg; rewrite S1_convn.
15 rewrite (bigID (fun i => g i \in X)) /=.
16 set b := \ssum_(i | _) -.
17 set c := \ssum_(i | _) -.
18 move=> zbc.
19 exists b; first exact: ssum_scaled_set.
20 exists c => //.
21 apply: (@ssum_scaled_set _ [pred i | g i \notin X]) => i /=.
22 move/asboolP; rewrite in_setE.
23 by case: (gT (g i) (imageP _ I)).
24 Qed.

```

6 Formalization of convex functions

In this section, we first (Sect. 6.1) formalize a generic definition of convex functions based on convex spaces; for that purpose, we introduce in particular *ordered convex spaces*. To demonstrate this formalization, we then apply it to the proof of the concavity of the logarithm function and to an information-theoretic function (Sect. 6.2).

6.1 Ordered convex spaces and convex functions

An ordered convex space extends a convex space with a partial order structure:

Definition 10 (**Module** `OrderedConvexSpace` in [18]). *An ordered convex space is a structure whose signature extends the one of convex spaces as follows:*

- Convex space X .
- Ordering relation $(\leq) \subset X \times X$.
- Reflexivity law: $x \leq x$.
- Transitivity law: $x \leq y \wedge y \leq z \Rightarrow x \leq z$.
- Antisymmetry law: $x \leq y \wedge y \leq x \Rightarrow x = y$.

The above definition does not force any interaction between convexity and ordering. It would also be a natural design to include an axiom stating that convex combinations preserve ordering [21, Sect. 2]. We however do not need such interactions for defining convex functions, which is our purpose here.

Convexity of a function is defined if its codomain is an ordered convex space. In the following, let T be a convex space and U be an ordered convex space.

Definition 11 (`convex_function_at` in [18]). *A function $f : T \rightarrow U$ is convex at $p \in [0, 1]$ and $x, y \in T$ if $f(x \triangleleft_p \triangleright y) \leq f(x) \triangleleft_p \triangleright f(y)$.*

Definition 12 (`convex_function` in [18]). *A function $f : T \rightarrow U$ is convex if it is convex at all $p \in [0, 1]$ and $x, y \in T$.*

The above predicates expect total functions. For partial functions, we resort to convex sets (Def. 8).

Definition 13 (`convex_function_in` in [18]). *Let D be a convex set in T . A function $f : T \rightarrow U$ is convex in D if it is convex at any $p \in [0, 1]$ and $x, y \in D$.*

Concave functions are defined similarly since f is concave for the order \leq if it is convex for \geq . When the codomain of f is \mathbb{R} , the prototypical example of an ordered convex space, it is also easy to prove that f is concave if $-f$ is convex.

6.2 Examples of convex functions

As a first example, we prove that the real logarithm function is concave. The concavity of logarithm is frequently used in information theory, for example, properties of data compression depend on it [4].

The definition of logarithm we use in COQ is the one of the standard library; it has the entire \mathbb{R} as its domain by setting $\log(x) = 0$ for $x \leq 0$. The statement of concavity is then restricted to the subset $\mathbb{R}_{>0}$.⁴

⁴ This way of restricting the domain of functions in their properties rather than in the definitions is a design choice often found in COQ. It makes it possible for functions such as the logarithm to be composable without being careful about their domains and ranges, and leads to a clean separation between definitions and properties of functions in the formalization.

Lemma 5 (`log_concave` in [17, `probability/ln_facts.v`]). *The extended logarithm function*

$$x \mapsto \begin{cases} \log(x) & \text{if } x \in \mathbb{R}_{>0} \\ 0 & \text{otherwise} \end{cases}$$

is concave in $\mathbb{R}_{>0}$.

The statement in COQ of these lemmas is as follows:

Lemma `log_concave` : `concave_function_in Rpos_interval log`.

The predicate `concave_function_in` has been explained in Sect. 6.1. The object `Rpos_interval` is the set of positive numbers described as the predicate `fun x => 0 < x` equipped with the proof that this set is indeed convex. The heart of the proof is the fact that a function whose second derivative is non-negative is convex (**Section** `twice_derivable_convex` in [18]). Our proof proceeds by using the formalization of real analysis from the COQ standard library; our formalization of convex spaces can thus be seen as an added abstraction layer of convexity to this library.

Our second example of convex function is the *divergence* (a.k.a. relative entropy or Kullback-Leibler divergence) of two probability distributions: an important information-theoretic function. Let `P` and `Q` be two finite distributions (over some finite type `A`). Their divergence `div` is defined as follows:

Variables (`A` : `finType`) (`P Q` : `fdist A`).

Definition `div` := `\sum_(a in A) P a * log (P a / Q a)`.

Actually, `div P Q` is defined only when `Q` *dominates* `P`, i.e., when `Q a = 0` implies `P a = 0` for all `a`. We call such a pair of probability distributions a *dominated pair*. Hereafter, we denote `div P Q` by `D(P || Q)` and the dominance of `P` by `Q` by `P <<< Q`.

We now show that the divergence function is convex over the set of dominated pairs. To formalize this statement using our definitions, we first need to show that dominated pairs form a convex space. To achieve this, it suffices to define the convex combination of the dominated pairs `a <<< b` and `c <<< d` as `a <| p |> c <<< b <| p |> d` (where we use the convex combination of probability distributions). This operator is easily shown to enjoy the properties of convex spaces (Sect. 2). Once this is done, one just needs to uncurry the divergence function to use the `convex_function` predicate:

Lemma `convex_div` : `convex_function (uncurry_dom_pair (@div A))`.

The proof follows the standard one [12, Thm. 2.7.2] and relies on the log-sum inequality formalized in previous work [6].

In previous work [5], we applied above results to the proofs of convexity of other information-theoretic functions such as the entropy and the mutual information.

7 Related work

Conical spaces have been known in the literature to work as a nice-behaving replacement of convex spaces when constructing models of nondeterministic computations. Varacca and Winskel [31] used convexity when building a categorical monad combining probability and nondeterminism, but they chose to avoid the problem of equational laws in convex spaces by instead working with conical spaces. There is a similar preference in the study of domain-theoretic semantics of nondeterminism, to a conical structure (d-cones [23]) over the corresponding convex structure (abstract probabilistic domain [20]). The problem is the same in this case: the difficulty in working with the equational laws of convex spaces [22, 30].

Flood [13] proposed to use conical spaces to investigate the properties of convex spaces. He showed that for any convex space, there is an enveloping conical space and the convex space is embedded in it. (A version of the embedding for convex sets into cones in vector spaces was already present in Semadini’s book [26].) Keimel and Plotkin [21] extended the idea for their version of ordered convex spaces and applied it in the proof of their key lemma [21, Lemma 2.8], which is an ordered version of the one proved by Neumann [25, Lemma 2].

Another aspect of convex spaces is the relationship to probabilistic distributions. From any set, one can freely generate a convex space by formally taking all finite convex combinations of elements of this set. The resulting convex space can be seen as a set of distributions over the original set, since the formal convex combinations are equivalent to distributions over the given points. By this construction, convex spaces serve as a foundation for the algebraic and category-theoretic treatments of probability. This allows for another application of our work to the semantics of probabilistic and nondeterministic programming [16, 19]. We have also been investigating this topic [3, 7]. Our most recent result [2] is based on the properties of convex sets and convex hulls, and deals with derived notions such as convex powersets. Its purpose is the formal study of program semantics from a category-theoretic point of view, rather than the formal study of the mathematical structure of convex spaces itself, which is rather the purpose of this paper.

8 Conclusion

In this paper, we formalized convex and conical spaces and developed their theories. In particular, we formally studied the various presentations of the convex combination operator, be it binary or multiary (Sect. 3). We provide formal proofs of the equivalence between several axiomatizations of both operators, where “proofs” in the literature were often only mere references to Stone’s foundational paper [27], while it only contains a reduction of the multiary case to the binary one. Based on convex and conical spaces, we also developed a theory of convex functions and of convex hulls. We illustrated these developments with detailed examples from real analysis and information theory.

Acknowledgments We acknowledge the support of the JSPS KAKENHI Grant Number 18H03204. We also thank Shinya Katsumata for his comments.

References

1. Affeldt, R., Cohen, C., Rouhling, D.: Formalization techniques for asymptotic reasoning in classical analysis. *J. Formaliz. Reason.* **11**(1), 43–76 (2018)
2. Affeldt, R., Garrigue, J., Nowak, D., Saikawa, T.: A trustful monad for axiomatic reasoning with probability and nondeterminism (March 2020), <https://arxiv.org/abs/2003.09993>
3. Affeldt, R., Garrigue, J., Nowak, D., Saikawa, T., Sauvage, C., Tanaka, K.: Monadic equational reasoning in Coq. <https://github.com/affeldt-aist/monae/> (2019), Coq scripts.
4. Affeldt, R., Garrigue, J., Saikawa, T.: Examples of formal proofs about data compression. In: *International Symposium on Information Theory and Its Applications (ISITA 2018)*, Singapore, October 28–31, 2018. pp. 665–669. IEICE. IEEE Xplore (Oct 2018)
5. Affeldt, R., Garrigue, J., Saikawa, T.: Reasoning with conditional probabilities and joint distributions in Coq. *Computer Software* (2020), https://staff.aist.go.jp/reynald.affeldt/documents/cproba_preprint.pdf, to appear. Japan Society for Software Science and Technology
6. Affeldt, R., Hagiwara, M., Sénizergues, J.: Formalization of Shannon’s theorems. *Journal of Automated Reasoning* **53**(1), 63–103 (2014)
7. Affeldt, R., Nowak, D., Saikawa, T.: A hierarchy of monadic effects for program verification using equational reasoning. In: Hutton, G. (ed.) *13th International Conference on Mathematics of Program Construction (MPC 2019)*, 7–9 October 2019, Porto, Portugal. pp. 226–254. Springer (2019)
8. Beaulieu, G.: Probabilistic completion of nondeterministic models. Ph.D. thesis, University of Ottawa (2008)
9. Bertot, Y., Gonthier, G., Ould Biha, S., Pasca, I.: Canonical big operators. In: Mohamed, O.A., Muñoz, C., Tahar, S. (eds.) *21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2008)*, Montreal, Canada, August 18–21, 2008. pp. 86–101. Springer (2008)
10. Bonchi, F., Silva, A., Sokolova, A.: The Power of Convex Algebras. In: Meyer, R., Nestmann, U. (eds.) *28th International Conference on Concurrency Theory (CONCUR 2017)*. Leibniz International Proceedings in Informatics (LIPIcs), vol. 85, pp. 23:1–23:18. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2017). <https://doi.org/10.4230/LIPIcs.CONCUR.2017.23>
11. Cheung, K.H.: Distributive Interaction of Algebraic Effects. Ph.D. thesis, University of Oxford (2017)
12. Cover, T.M., Thomas, J.A.: *Elements of information theory*. Wiley (2006), 2nd ed.
13. Flood, J.: Semiconvex geometry. *Journal of the Australian Mathematical Society* **30**(4), 496–510 (1981). <https://doi.org/10.1017/S1446788700017973>
14. Fritz, T.: Convex spaces I: Definition and examples (2015), available at <https://arxiv.org/abs/0903.5522>. First version: 2009
15. Garillot, F., Gonthier, G., Mahboubi, A., Rideau, L.: Packaging mathematical structures. In: *22nd Int. Conf. on Theorem Proving in Higher Order Logics (TPHOLs 2009)*, Munich, Germany, August 17–20, 2009. *Lecture Notes in Computer Science*, vol. 5674, pp. 327–342. Springer (2009)

16. van Heerdt, G., Hsu, J., Ouaknine, J., Silva, A.: Convex language semantics for nondeterministic probabilistic automata. In: Fischer, B., Uustalu, T. (eds.) 15th International Colloquium on Theoretical Aspects of Computing (ICTAC 2018), 12–19 October 2018, Stellenbosch, South Africa. pp. 472–492. Springer (2018)
17. Infotheo: A Coq formalization of information theory and linear error-correcting codes. <https://github.com/affeldt-aist/infotheo/> (2020), Coq scripts.
18. Infotheo: `probability/convex.choice.v`. In: [17] (2020), Coq scripts.
19. Jacobs, B.: Convexity, duality and effects. In: IFIP TCS. IFIP Advances in Information and Communication Technology, vol. 323, pp. 1–19. Springer (2010)
20. Jones, C., Plotkin, G.D.: A probabilistic powerdomain of evaluations. In: [1989] Proceedings. Fourth Annual Symposium on Logic in Computer Science. pp. 186–195 (June 1989). <https://doi.org/10.1109/LICS.1989.39173>
21. Keimel, K., Plotkin, G.: Mixed powerdomains for probability and non-determinism. *Logical Methods in Computer Science* **13** (12) (2016). [https://doi.org/10.23638/LMCS-13\(1:2\)2017](https://doi.org/10.23638/LMCS-13(1:2)2017)
22. Keimel, K., Plotkin, G.D.: Predicate transformers for extended probability and non-determinism. *Mathematical Structures in Computer Science* **19**(3), 501–539 (2009). <https://doi.org/10.1017/S0960129509007555>
23. Kirch, O.: *Bereiche und Bewertungen*. Master’s thesis, Technischen Hochschule Darmstadt (1993)
24. Mahboubi, A., Tassi, E.: Canonical structures for the working Coq user. In: 4th International Conference on Interactive Theorem Proving (ITP 2013), Rennes, France, July 22–26, 2013. *Lecture Notes in Computer Science*, vol. 7998, pp. 19–34. Springer (2013)
25. Neumann, W.D.: On the quasivariety of convex subsets of affine spaces. *Archiv der Mathematik* **21**, 11–16 (1970)
26. Semadini, Z.: *Banach Spaces of Continuous Functions*. PWN (1971)
27. Stone, M.H.: Postulates for the barycentric calculus. *Ann. Mat. Pura Appl.* **29**(1), 25–30 (1949)
28. Świrszcz, T.: Monadic functors and convexity. *Bulletin de l’Académie polonaise des sciences. Série des sciences mathématiques, astronomiques et physiques* **22**(1) (1974)
29. The Coq Development Team: *The Coq Proof Assistant Reference Manual*. Inria (2019), available at <https://coq.inria.fr>. Version 8.11.0
30. Tix, R., Keimel, K., Plotkin, G.: Semantic domains for combining probability and non-determinism. *Electronic Notes in Theoretical Computer Science* **222**, 3 – 99 (2009). <https://doi.org/https://doi.org/10.1016/j.entcs.2009.01.002>
31. Varacca, D., Winskel, G.: Distributing probability over non-determinism. *Mathematical Structures in Computer Science* **16**(1), 87–113 (2006)