Ergodic Average in Constraint Programming (Extended Abstract)

Alessandra Di Pierro
dipierro@di.unipi.it
Dipartimento di Informatica, Universitá di Pisa, Italy

Herbert Wiklicky
h.wiklicky@doc.ic.ac.uk
Department of Computing, Imperial College, London, UK

July 22, 1999

1 Introduction

We will discuss the problem of modelling probabilistic properties of constraint programs, which express the average of some quantities of interest. A random variable on the set of constraints is used for assigning to each constraint a real value representing the 'cost' of that constraint. This way, we obtain a notion of *quantitative observables* Q which, although interesting in itself can be used in order to define different types of *average properties*. One, $E(Q)$ refers to the average over different runs of a probabilistic programs and corresponds to the expected value of the random variable. Another one, $A(Q)$, considers the average value of the quantity along the same (infinite) program run. In this work we concentrate on the second type of properties, which are also called *long-run average properties* [2].

We consider here the Probabilistic Concurrent Constraint Programming (PCCP) language which has been previously introduced by the authors in [4, 5]. For this language a denotational semantics was presented which is based on a Banach space construction on the underlying constraint system $\mathcal{C}$. Following the standard approach in denotational semantics, we define a fixed point operator Φ for assigning a denotation to PCCP programs. A fixed point $I^*(A) = \Phi(I^*(A))$ of Φ is then taken as the denotation of an agent A [5]. With respect to the corresponding measure μ_{I^*} the operator Φ is measure-preserving. This allows us to exploit a fundamental theorem in ergodic theory to obtain a characterisation of the long-run average $A(Q)$ of a quantitative observable Q.

2 Observing Probabilistic Systems

Given a probabilistic system, be it a natural phenomena or the execution of a probabilistic program, we can look at different aspects of its behaviour. We can, for example, ask what will be the *final* result or we can ask what is happening *during* the process. Theoretical work is, of course, not on the observation itself but on forecasting the observation, i.e. given the specification of the system, what can we expect to observe. To illustrate this look at the following two situations.

2.1 Final Observation

[???? A Good Example ????]

This kind of observation of a process is typically, for example, with finite computations, as one is simply interested in what situation to expect at the end of a process. But there are also infinite processes/computations where this type of analysis is relevant (e.g. counting).

2.2 Average Observation

Consider another example. Suppose we have a printer which is controlled by a driver or spooler. Let us assume that we have two printing modes (e.g. colour and black & white) with different costs (e.g. expensive and cheap).

We can then define a function which associates to each state $s \in S$ of the printer the corresponding cost $Q(s)$. An example of such a (static) cost function $Q : S \mapsto \mathbb{R}$ is given by:

$$Q(s) = \begin{cases} 0 & \text{iff} \quad \text{printer is not operating} \\ 1 & \text{iff} \quad \text{black \& white printing} \\ 5 & \text{iff} \quad \text{colour printing} \end{cases}$$

Then one can ask, how much money is spent in the average if the printer is operating for a long time. This can be represented by an appropriate *average*.

3 Probabilistic Constraint Concurrent Programming

Probabilistic Concurrent Constraint Programming was introduced in [4], to the purpose of formalising randomised algorithms.

In this language randomness is expressed in the form of a *probabilistic choice*, which replaces the non-deterministic choice of CCP and allows a program to make stochastic moves during its execution. This construct also allows us to introduce the element of chance directly at the algorithmic level without the need of any modification at the data level. Thus, the constraint system $\mathcal{C}$ underlying the language does not need to be re-structured according to some probabilistic or fuzzy or belief system, and we can assume the definition as a cylindric algebraic cpo given in [7], to which we refer for more details.

The syntax of the PCCP *agents* is given by the grammar:

$$A ::= \mathbf{stop} \;\big|\; \mathbf{tell}(c) \;\big|\; \textstyle\fbox{}_{i=1}^{n}\; c_i \mid p_i \to A_i \;\big|\; A \parallel A \;\big|\; \exists_x A \;\big|\; p(x)$$

where c and c_i are *finite* constraints in $\mathcal{C}$. A PCCP process P is then an object of the form $D.A$, where D are procedure declarations of the form $p(x) : -A$.

3.1 Operational Semantics

PCCP agents are the same as CCP agents but for the non-deterministic choice construct, which is replaced by the probabilistic one, $\fbox{}_{i=1}^{n}\; c_i \mid p_i \to A_i$. The operational meaning of this construct is as follows: First, check whether constraints c_i are entailed by the store. Then we have to *normalise* the probability distribution by considering only the enabled agents. This can be done by considering for enabled agents the normalised transition probability, $\tilde{p}_i = \frac{p_i}{\sum_{\vdash c_j} p_j}$, where the sum is over all enabled agents. Finally, one of the enabled agents is chosen according to this new probability distribution $\tilde{p}_i$.

$$
\begin{array}{lll}
\textbf{R1} & \langle \mathbf{tell}(c), d\rangle \longrightarrow_1 \langle \mathbf{stop}, c \sqcup d\rangle & \\
\textbf{R2} & \langle []_{i=1}^{n} c_i \mid p_i \to A_i, d\rangle \longrightarrow_{\tilde{p}_j} \langle A_j, d\rangle & j \in [1,n];\ d \vdash c_j \\
\textbf{R3} & \dfrac{\langle A, c\rangle \longrightarrow_p \langle A', c'\rangle}{\begin{array}{c}\langle A \parallel B, c\rangle \longrightarrow_p \langle A' \parallel B, c'\rangle \\ \langle B \parallel A, c\rangle \longrightarrow_p \langle B \parallel A', c'\rangle\end{array}} & \\
\textbf{R4} & \dfrac{\langle A, d \sqcup \exists_x c\rangle \longrightarrow_p \langle B, d'\rangle}{\langle \exists_x^d A, c\rangle \longrightarrow_p \left\langle \exists_x^{d'} B, c \sqcup \exists_x d'\right\rangle} & \\
\textbf{R5} & \langle p(y), c\rangle \longrightarrow_1 \langle A, c\rangle & \text{with } p(x) : -A \in P
\end{array}
$$

Table 1: The transition system for PCCP.

The operational semantics of PCCP – for details see [4] – is given in terms of a probabilistic transition system, $(Conf, \longrightarrow_p)$, where $Conf$ is the set of configurations $\langle A, d\rangle$ representing the state of the system at a certain moment, namely the agent A which has still to be executed and the common store d; and the transition relation $\longrightarrow_p$ is defined in Table 1.

3.2 Denotational Semantics

Whereas in the classical CCP semantics one is interested in what (possible) results, i.e. sets of constraints, the execution of a program leads to, the semantics of PCCP – cf. [5] – requires to consider not just sets of constraints as possible interpretations, but instead random distributions on the set of constraints in order to record also the probabilities of certain constraints to be computed.

Definition 1 *An interpretation $I(A)$ of an agent A is a probability distribution on $\mathcal{C}$, i.e. $I(A) : \mathcal{C} \mapsto [0,1]$ with $\sum_{c \in \mathcal{C}} = 1$.*

For finite and countable constraint systems interpretations can simply be represented by vectors in $l^1(\mathcal{C})$. We will concentrate here on this setting; it is nevertheless possible to extend our approach also to uncountable constraint systems by considering the appropriate measure theoretic framework.

To keep the presentation even simpler we will consider in the following a sub-language of PCCP – we sometimes refer to as PCLP [5] – where all guards are *true*. This is to avoid various problems related to concurrency (e.g. synchronisation) for the time being.

The denotational semantics of an agent A is defined in terms of a fixed-point operator Φ on the set of interpretations $\mathcal{I}$. The definition of Φ is given in Table 2 (for details see [5, 3]).

$\Phi(I)(\mathbf{stop})$	$=$	$\{\langle true, 1\rangle\}$
$\Phi(I)(\mathbf{tell}(c))$	$=$	$\{\langle c, 1\rangle\}$
$\Phi(I)(\square_{i=1}^{n}\ true \mid p_i\ \rightarrow\ A_i)$	$=$	$\sum_{i=1}^{n}\ p_i \cdot \Phi(I)(A_i)$
$\Phi(I)(A_1 \parallel A_2)$	$=$	$\Phi(I)(A_1)\ \otimes\ \Phi(I)(A_2)$
$\Phi(I)(\exists_x A)$	$=$	$\exists_x \Phi(I)(A)$
$\Phi(I)(p(x))$	$=$	$I(A)$ with $p(x) : -A \in P$

Table 2: The compositional definition of $\Phi : \mathcal{I} \rightarrow \mathcal{I}$.

4 Observing the Behaviour

4.1 Standard I/O Observables

The transition system in Table 1 describes how a computation proceeds operationally. The semantics of a program is usually defined in terms of the final result of this process. Depending on what one is interested in one may consider different *observables*.

In PCCP the standard I/O observables are based on the following notion of the results R of a program P starting with an initial store d. This is the set of all pairs $\langle c, p\rangle$, where c is the least upper bound of the partial constraints accumulated during a computation starting from d; and p is the probability of reaching that result.

$$\begin{array}{rcl} \mathcal{R}(A, d) & = & \{\langle c, p\rangle \mid \langle A, d\rangle \longrightarrow_p^* \langle A_1, c\rangle \not\longrightarrow\} \cup \\ & & \{\langle \bigsqcup_i d_i, \prod_i p_i\rangle \mid \langle A, d_0\rangle \longrightarrow_{p_0} \ldots\}. \end{array}$$

The first term describes the results of finite computations where the least upper bound of the partial store corresponds to the final store. The second term covers infinite computation. By appropriately normalising $\mathcal{R}$, we obtain a notion of observables $\mathcal{O}(A, d)$, corresponding to the results of executing A from store d together with their probability (see [4]).

4.2 Average and Costs

However, for analysis purposes we might need to consider further quantitative information, e.g. costs, which are not provided by the probabilistic observables. Such information can be modelled by means of random variables.

Definition 2 *Let $\mathcal{C}$ denote the set of constraints and A an agent. A quantitative observable Q_A is a random variable. i.e a real-valued mapping $Q_A : \mathcal{C} \mapsto \mathbb{R}$.*

Although these quantities are of importance in their own right, our interest is in the average behaviour of a program w.r.t. such quantities. It is important to distinguish between two different types of averages.

On one hand, we can look at the average "cost" over several executions of the program. For example consider the different executions of a PCCP program A starting from a given store d. The I/O observables provides us with a 'final' distribution assigning a probability to each result. Then suppose that we are able to estimate some cost associated to each store, and let Q_A be a random variable expressing such information. The average cost is given by the expected value of Q_A wrt the initial store d,

$$E(Q_A, d) = \sum_{\langle c,p\rangle \in \mathcal{O}(A,d)} Q(c) \cdot p$$

(we are assuming that the distribution of Q and the final observable distribution coincide).

On the other hand, we can also ask for the average value of a certain quantity along a computational path $\langle A, d_0\rangle \longrightarrow_{p_0} \langle A_1, d_1\rangle \longrightarrow_{p_1} \ldots$. This leads to the definition of the long-run average as:

$$A(Q_A, d) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Q_A(d_i) Prob(d_i),$$

where $Prob(d_i)$ is the partial distribution, i.e. the distribution of the partial results after i steps.

Note that this notion is only relevant to infinite computations. For terminating computations one had to extend the computational path such that the final configuration would be repeated (infinitely) often, but then the average would be simply dominated by the probability distribution of this terminal configuration.

5 Fixpoints and Observables

Taking the denotational semantics, i.e. the fixpoint operator Φ, we notice that we can look at different types of fixpoints which correspond to different types of observations. In [3] we identified various candidates, tow of them are of interest in the current context.

Note, that if we are looking for a fixpoint of Φ then in terms of measures and distributions on $\mathcal{C}$ we are looking for a Φ-invariant measure I^* on $\mathcal{C}$.

5.1 Pointwise Fixed Point

A first method to construct a fixed-point of Φ is simply via the iterative application of Φ (staring with an extremal interpretation like $I_0 = \{\langle true, 1\rangle\}$) and by constructing the pointwise limit of $I_n = \Phi^n I_0$. The latter is defined for a sequence of distributions $I_n = \{\langle c_i, p_{i,n}\rangle\}$ as:

$$I_p^* = \lim_{n\to\infty} I_n = \left\{ \left\langle c_i, \lim_{n\to\infty} p_{i,n} \right\rangle \right\}_i .$$

If the limit exists then, it is a fixed-point of Φ because of the continuity of Φ. The convergence of this method was investigated in [5].

5.2 Ergodic Fixed Point

Another approach would be to look at the average interpretation (instead of a pointwise limit) occurring during the computation. We then had to construct the limit of interpretations in the average:

$$I_e^* = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} I_j = \left\{ \left\langle c_i, \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} p_{i,j} \right\rangle \right\}_i .$$

In this case we could construct a fixed-point using some "Ergodic Theorem" (see [9, Theorem VIII.3.2,p 213], [6, p 232], or [8]) to guarantee the convergence. For example:

Theorem 1 *(Nonlinear Mean Ergodic Theorem) Let K be a bound, closed, convex subset of a uniformly convex Banach space X with Frechet differential norm, and let $\Phi : K \mapsto K$ be a non-expansive map. For $x \in K$, let*

$$S_n(x) = \frac{1}{n} \sum_{i=0}^{n-1} \Phi^i(x).$$

Then $\{S_n(x)\}$ converges weakly to a fixed point of Φ.

6 Calculation of Averages

Ergodic theorems guarantee existence of invariant measures, i.e. fixpoints, but we can use the ergodic fixpoint to calculate the average of some quantitative properties.

Given a cost function $Q : \mathcal{C} \mapsto \mathbb{R}$. Construct the ergodic fixpoint I_e^* then the expected value for Q with respect to I_e^* is the average:

$$A(Q) = E_e(Q)$$

The idea is thus: The ergodic fixpoint describes the average probability that a constraint c is in the store during an (infinite) computation, so to say how often the computation visits c. Once we know this probability, i.e. the ergodic fixpoint distribution I_e^*, we can immediately calculate the average $A(Q)$ for *different* (static) cost function by the above formula.

For finite constraint systems the argument is nearly trivial. If we look at the average of a Q we have to calculate its expected cost in each computational step, obviously if I_n describes the distribution (front) in the n-th step, then the expected cost of Q is:

$$E_n(Q) = \sum_{c\in\mathcal{C}} I_n(c)Q(c).$$

In order to get the average during the computation, we have to calculate the average of E_n:

$$A(Q) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E_n(Q) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{c\in\mathcal{C}} I_n(c)Q(c).$$

As everything is finite and (therefore) continuous, this is the same as:

$$A(Q) = \lim_{n\to\infty} \frac{1}{n} \sum_{c\in\mathcal{C}} \sum_{i=1}^{n} I_n(c)Q(c) = \sum_{c\in\mathcal{C}} \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_n(c)Q(c).$$

Figure 1: The development of the distributions.

The ergodic fixpoint or average observables (whose existence was guaranteed by the ergodic theorem even for the general case of an infinite constraint system) was exactly defined as:

$$I_e^*(c) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_n(c).$$

Therefore, we get the following expression for the average of Q:

$$A(Q) = \sum_{c\in\mathcal{C}} \left(\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_n(c) \right) Q(c) = \sum_{c\in\mathcal{C}} I_e^*(c)Q(c)$$

or all together:

$$A(Q) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E_n(Q) = \sum_{c\in\mathcal{C}} I_e^*(c)Q(c) = E_e(Q)$$

The situation is depicted in Figure 1. What is basically happening is that we replace the sum (integral) over time and the sum (integral) over the set of constraints.

7 Further Work

The purpose of the work sketched here is to relate the analysis of average properties of programs to the classical framework of ergodic theory. Various topics in this setting still need further investigation.

Concretely, we would like to gain a better understanding of the relation between different average constructions, for example when the average is computed using different (but equivalent) semantic operators. It would also be interesting to investigate the relation between $E(Q)$ and $A(Q)$.

Further work should also be concentrated on the use of more general ergodic theorems [1]. In particular the von Neumann Mean Ergodic Theorem and similar results could provide a more abstract formulation in terms of operators and elements of a Hilbert or Banach space.

References

[1] Jon Aaronson. *An Introduction to Infinite Ergodic Theory.* American Mathematical Society, Providence, Rhode Island, 1997.

[2] Luca de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, Department of Computer Science, 1998.

[3] Alessandra Di Pierro and Herbert Wiklicky. A Banach space based semantics for Probabilistic Concurrent Constraint Programming. In *CATS'98*, volume 20(3) of *Australian Computer Science Communications*, pages 245–259, Singapore – Berlin – New York, 1998. Springer Verlag.

[4] Alessandra Di Pierro and Herbert Wiklicky. An operational semantics for Probabilistic Concurrent Constraint Programming. In *ICCL'98*, pages 174–183. IEEE Computer Society Press, 1998.

[5] Alessandra Di Pierro and Herbert Wiklicky. Probabilistic Concurrent Constraint Programming: Towards a fully abstract model. In *MFCS'98*, volume 1450 of *Lecture Notes in Computer Science*, pages 446–455, Berlin – New York, August 1998. Springer Verlag.

[6] Kazimierz Goebel and W.A. Kirk. *Topics in Metric Fixed Point Theory*, volume 28 of *Cambridge studies in advanced mathematics.* Cambridge University Press, Cambridge, 1990.

[7] Vijay A. Saraswat, Martin Rinard, and Prakash Panangaden. Semantic foundations of concurrent constraint programming. In *18th POPL*, pages 333–52. ACM SIGPLAN/SIGACT, 1991.

[8] Peter Walters. *An Introduction to Ergodic Theory*, volume 79 of *Graduate Texts in Mathematics.* Springer Verlag, New York – Berlin – Heidelberg, 1982.

[9] Kôsaku Yosida. *Functional Analysis.* Springer Verlag, Berlin – Heidelberg – New York, 1980.

A An Ergodic Theorem

[An alternative to the general ergodic theorem from CATS, need reformulation and comparison]

There exist a number of so-called ergodic theorems, e.g. Birkhoff, von Neumann, Hopf etc. [8, 1]. We will look at them in order to guarantee the existence of certain long-run averages $A(Q)$ of quantitative observables Q.

The following notion is introduced in [**?**, Def.1.3]:

Definition 3 *A linear operator T defined for functions on X is* doubly stochastic *if it maps L^1 into L^1 and satisfies for all $f \in L^1$ the following conditions:*

1. *$f \geq 0$ implies $Tf \geq 0$*
2. *$\int_X Tf dm = \int_X f dm$*
3. *$T1 = 1$*

Assume that $(X, \mathcal{B}, m)$ is a normalised finite measure space, then the Yosida Mean Ergodic Theorem is stated as follows [**?**, Thm.1.4]:

Theorem 2 *Let T be a doubly stochastic operator and $f \in L^p$. Then there exists $f^* \in L^p$ such that:*

$$\lim_{n \to \infty} \left\| \frac{1}{n} \sum_{i=0}^{n-1} T^i f - f^* \right\|_p = 0.$$

For a finite or countable constraint system $\mathcal{C}$, we take the powerset $\mathcal{P}(\mathcal{C})$ as the σ-algebra of measurable sets; an interpretation I, i.e. a random distribution, gives us a measure on this space via $m_I(B) = \sum_{c \in B} I(c)$. Note, that in this case we can write the integrals as simple sums, e.g. $\int f dm_I$ becomes in our case $\sum_{c \in \mathcal{C}} Q(c)I(c)$.

If we take as T the fixed point operator Φ we see that the fixed point interpretation $I^* = \Phi(I^*)$ gives us a corresponding invariant measure. Furthermore, if we take as f a cost function Q which is in $L^1(\mathcal{C}) = l^1(\mathcal{C})$, i.e. $\sum_{c \in \mathcal{C}} |Q(c)|I^*(c) < \infty$, then the Ergodic Theorem guarantees the existence of the long-run average (cost) function $Q^* = A(Q)$.

A Long Run Average: An Example

In order to exemplify our approach and to illustrate the concept of a long-run average we present a nearly trivial example.

This simple example to be discussed can be represented in form of a legend: A father had tow sons and an precious diamond ring. On his death he decided that to avoid fighting between the two brothers, each should have the ring for one year. The argument of the father then was that each of the brothers would own half the ring.

To formalise this, let us look at the following trivial (deterministic) counting program. It does not terminate but adds at each computational steps a new constraint $x_n = n$ to the store (if we start with calling $count(0)$).

$$count(x) \quad :- \quad \exists_y(\mathbf{tell}(y = x + 1) \parallel count(y))$$

We start as usual with $c_0 = true$ and get a sequence of stores: $c_1 = (x_1 = 1)$, $c_2 = (x_1 = 1 \wedge x_2 = 2)$, etc., i.e. point distributions I_n with $I_n(c_n) = 1$ and 0 for all other constraints.

Now we define a cost function or quantitative observable Q, such that

$$Q(c) = \begin{cases} 1 & \text{if } c = c_{2n} \\ 0 & \text{otherwise} \end{cases}$$

i.e. the "cost" of most constraints is 0; only if the store corresponds exactly to an even counting step $Q(c)$ is 1. This would correspond to the "ownership" of one of the two brothers.

Now we can ask for the "average ownership" (where we identify the point distribution I_n and the single constraint c_n,

$$\begin{aligned} A_n(Q)(c_0) &= \frac{1}{n}\sum_{i=1}^{n} Q(\Phi^n(count(0))(c_0)) = \\ &= \frac{1}{n}\sum_{i=1}^{n} Q(I_n) = \\ &= \frac{1}{n}\sum_{i=1}^{n} Q(c_n) \end{aligned}$$

and its long-run behaviour, i.e. $A(Q) = \lim A_n(Q)$. We get

$$A_n(Q)(true) = \begin{cases} \frac{1}{2} & \text{if } n = 2k \\ \frac{k}{2k+1} & \text{if } n = 2k+1 \end{cases}$$

and therefore the limit of these partial averages is

$$Q^*(true) = A(Q)(true) = \frac{1}{2}$$

Although the costs are oscillating and the overall costs $\sum Q(I_n)$ diverge, the average cost, i.e. the "average period of ownership" converges to a half – just as the father had intended it.

A Ergodic Theorems

There exist a number of so-called ergodic theorems, e.g. Birkhoff, von Neumann, Hopf etc. [8, 1] wich basically all gurantee the existence of some average construction.

A.1 Nonlinear Mean Ergodic Theorem

In our situation we apply general theorems to guarntee the existence of the an average fixpoint even when more concrete ones might already be sufficient. The most general one is perhaps the following [6, p 232]:

Theorem 3 *(Nonlinear Mean Ergodic Theorem) Let K be a bound, closed, convex subset of a uniformly convex Banach space X with Frechet differential norm, and let $\Phi : K \mapsto K$ be a non-expansive map. For $x \in K$, let*

$$S_n(x) = \frac{1}{n} \sum_{i=0}^{n-1} \Phi^i(x).$$

Then $\{S_n(x)\}$ converges weakly to a fixed point of Φ.

In order to show the applicability of this theorem we have to look at the properties of our concrete situation.

A.1.1 The Space $(l^1(\mathcal{C}), \|.\|_1)$

The arena of our construction is the (real) Banach space $(l^1(\mathcal{C}), \|.\|_1)$, it can be defined to be the set of tuples or countable sequnces of real numbers indexed by elements of the (finite or countable) set of constraints $\mathcal{C}$ such that the 1-norm exists.

For a sequence (or tuple) of real numbers (indexed by elements of $\mathcal{C}$ we define the (1-)norm:

$$\|(x_c)_{c\in\mathcal{C}}\| = \|(x_c)_{c\in\mathcal{C}}\|_1 = \sum_{c\in\mathcal{C}} |x_c|$$

and

$$l^1(\mathcal{C}) = \left\{ (x_c)_{c\in\mathcal{C}} \mid \sum_{c\in\mathcal{C}} |x_c| < \infty \right\}$$

This space, equiped with the pointwise addition and scalar multiplication has the following properties:

- $l^1(\mathcal{C})$ is a vector space.
- $(l^1(\mathcal{C}), \|.\|_1)$ is a normed vector space.
- $(l^1(\mathcal{C}), \|.\|_1)$ is a Banach space.

because all Cauchy sequences converge.

Definition 4 *[9, p126] A normed linear space is called* uniformly convex *if*

$$\forall \varepsilon > 0, \exists \delta > 0 : \|x\| \leq 1, \|y\| \leq 1 \text{ and } \varepsilon \leq \|x - y\| \Rightarrow \|x + y\| \leq 2(1 - \delta)$$

All Pre-Hilbert spaces and the spaces l^p and L^p for $1 < p < \infty$ are uniformly convex [9, p127]. For a finite $\mathcal{C}$ nevertheless we still have:

- $(l^1(\mathcal{C}), \|.\|_1)$ is uniformly convex.

Definition 5 *[?, p458] Let X and Y be normed linear spaces, $x_0 \in X$ be a fixed point, and $f : X \mapsto Y$. We say that f has a* Fréhet derivative *at x_0 if there exists a bounded linear operator $F : X \mapsto Y$ such that:*

$$\lim_{\|h\| \to 0} \frac{\|f(x_0 + h) - f(x_0) - F(h)\|}{\|h\|} = 0$$

To be shown therefor is the following:

- $\|.\| : l^1(\mathcal{C}) \mapsto \mathbb{R}$ has a *Fréhet derivative.*

A.1.2 The Simplex $\mathcal{S}$

The important fact is that Φ leaves the simplex

$$\mathcal{S} = \left\{ (x_c)_{c \in \mathcal{C}} \in l^1(\mathcal{C}) \mid x_c > 0 \text{ and } \sum_{c \in \mathcal{C}} x_c = 1 \right\}$$

i.e. the set of interpretations invariant.

Definition 6 *Let $\S \subseteq X$ be a subset of a normed linear space (over the reals) X.*
Then S is called bounded *if*

$$\sup_{s \in S} \|s\| < infty.$$

The set is called convex *if $\forall s, t \in S$ and $0 \leq \lambda \leq 1$*

$$\lambda s + (1 - \lambda)t \in S.$$

For the simplex the following properties hold:

- $\mathcal{S}$ is bounded.
- $\mathcal{S}$ is closed.
- $\mathcal{S}$ is convex.

A.1.3 The operator Φ

Definition 7 *Let $\Phi : X \mapsto X$ be a mappring from a normed linear space (over the reals) X into itself.*
Then Φ is called a linear *operator if $\forall x, y \in X$ and $\alpha \in \mathbb{R}$:*

$$\Phi(x + y) = \Phi(x) + \Phi(y) \text{ and } \Phi(\alpha x) = \alpha \Phi(x).$$

The operator Φ is called bounded *if $\forall x \in X$:*

$$\sup_{\|x\|=1} \|\Phi(x)\| < \infty.$$

The operator Φ is called non-expansive *if $\forall x, y \in X$*

$$\|\Phi(x) - \Phi(y)\| \leq \|x - y\|.$$

The last property actually applies already to metric spaces: $d(\Phi(x), \Phi(y)) \leq d(x, y)$.

It is well known that the first two properties are relatively strong as for linear operators the following holds: Φ is continuous iff Φ is bounded (e.g. [9, p43]).

In our case with the fixpoint operator Φ on $l^1(\mathcal{C})$ the situation is simpel because Φ maps the simplex $\mathcal{S}$ and the unit sphere linearly into itself. Therefore, the following properties hold for our Φ:

- Φ is linear.
- Φ is bounded.
- Φ is continuous.
- Φ is non-expansive.