A Logic with Measurable Spaces for Natural Language Semantics*

Jean-Philippe Bernardy, Rasmus Blanck, and Aleksandre Maskharashvili

University of Gothenburg, Department of Philosophy, Linguistics and Theory of Science, Centre for Linguistics and Studies in Probability
firstname.lastname@gu.se

1 Introduction

The ability of humans to reason under uncertainty has reflections within natural language where we find various lexico-syntactic constructions which allow us to express uncertain information. Moreover, we are able draw conclusions - make inferences under uncertainty. To give an adequate account to this crucial aspect of natural language, it has been long argued for employing probabilistic tools in defining semantics of natural language. In this abstract we address this issue by proposing a Logic with Measurable Spaces (LMS). We argue that LMS is suitable to represent the semantics of a number of important natural language phenomena. LMS draws inspiration from several sources. It is decidable (like descriptive logics). It features Sigma spaces (like Martin-Löf type-theory). It internalises the notion of the cardinality (in fact, here, measures) of spaces (see Fox and Lappin [2]) and ratios thereof, allowing to capture the notion of event probability.

2 Syntax

The syntax of LMS comprises three categories: spaces (hereafter ranged over by A, B, C, etc.), propositions (ranged over by ϕ, ψ) and real-valued expressions (ranged over by e_i). We have four ways to construct spaces: (1) $\mathsf{Distr}(d)$, where d is any given continuous distribution over $\mathbb{R}$. (2) $\mathsf{IsTrue}(\phi)$, which is inhabited (by a single object $\diamond$) if ϕ is true or empty otherwise. (3) $\Sigma(x : A)B$, which generalises the pair space $A \times B$ when B depends on a value x of space A. If B is the space of proofs $\mathsf{IsTrue}(\phi)$, then $\Sigma(x : A)B$ is isomorphic to the subspace of A where ϕ holds. (4) $f\langle A \rangle$, which can be understood as the image of A under f.

Propositions can be constructed using boolean operators (such as in $\phi \wedge \psi$) and by comparison operators ($e_1 \leq e_2$). The syntax for expressions is mostly standard: we have variables, constants, projections, arithmetic expressions (e.g. $e_1 + e_2$). An exception is the expression $\mathsf{measure}(A)$, which represents the measure of a space. In our syntax a proof object is written as $\diamond$.

Expressions are typed – but their typing rules are entirely straightforward, thus we omit them in this extended abstract.

3 Semantics

Semantics are given by an evaluation function (written $[\![\cdot]\!]$), which maps propositions to a boolean, expressions to a real number and spaces to integrators. The semantics of expressions

* Supported by Swedish Research Council, Grant number 2014-39

and proofs is entirely straightforward, except for the measures of a space. In that case, we integrate over the whole space the constant value 1 — thus "counting" the elements of that space: $[\![\mathsf{measure}(A)]\!] = [\![A]\!](\lambda x.1)$.

Indeed, the semantics of spaces is parameterized over an additional function (f below), which is integrated over the whole space. Integrating over a distribution space proceeds in the usual way for a distribution. The measure of a proof space is 0 if the underlying proposition is false. Otherwise, by convention its measure is 1, and thus the integrator reduces to the integrated function at $\diamond$. Integration over a sigma space $\Sigma(x : A)B$ is the integration of B for all possible values x of A. Integration over a morphism works by composition. (If the morphism is not injective, the density of values in the image increases.)

$$[\![\mathsf{Distr}(d)]\!]f = \int_{\mathbb{R}} \mathrm{PDF}(d, x) \cdot f(x)dx \qquad [\![\mathsf{IsTrue}(\phi)]\!]f = \mathsf{if}\ [\![\phi]\!]\ \mathsf{then}\ f(\diamond)\ \mathsf{else}\ 0$$
$$[\![\Sigma(x : A)B]\!]f = [\![A]\!](\lambda x.[\![B]\!](\lambda y.f(x, y))) \qquad [\![g\wr A\wr]\!]f = [\![A]\!](f \circ g)$$

4 Meanings

4.1 Encoding quantifiers

Even though it has very few constructions, LMS is sufficently expressive to encode the usual logical quantifiers: every x in A satisfies ϕ iff the subspace of A where ϕ holds is (at least) as big as A itself.[1] The definition of the existential quantifier follows a similar pattern.

$$\forall x : A.\phi \triangleq \mathsf{measure}(A) \leq \mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi))$$
$$\exists x : A.\phi \triangleq 0 \leq \mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi))$$

4.2 Events and probabilities

A conditional event P can represented by a proposition ϕ, depending on a random variable x over a space A. (The space of x can be a Σ-space and thus stand for a vector of variables). The probability of P is given by[2]

$$\frac{\mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi))}{\mathsf{measure}(A)}$$

If P and Q are events represented respectively by propositions ϕ and ψ, and depending on a random variable x in space A, then the conditional probability P given Q is

$$\frac{\mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi \wedge \psi))}{\mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi))}$$

Given this, we can encode higher-order quantifiers (in the sense of Barwise and Cooper [1]), as follows:

$$\text{most } A \text{ satisfy } \phi \triangleq \frac{\mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi \wedge \psi))}{\mathsf{measure}(\Sigma(x : A).\mathsf{IsTrue}(\phi))} \geq \theta$$

for a suitable threshold θ.

[1] This definition is problematic when ϕ logically false for some x, but still stochastically true. To deal with such cases, one must then use disintegrators, as explained by Shan and Ramsey [3].

[2] This simplified formula will not work when A is stochastially empty. Again, one should use disintegrators.

4.3 Encoding predicates

LMS has no ostensible way to quantify over functions, so one might wonder how it can represent an unknown predicate. Quantifying over *all possible* predicates is of course out of the question — this can only be done in undecidable logics. While such a powerful operation is crucial for the foundations of mathematics, we find it to be much less important in linguistics. Indeed, we can still quantify over measurable subsets of predicates — which can be large enough to be useful. One such interesting subset is given by a positive cosine-distance correlation between word-vectors, which has gained considerable traction in lexical semantics in the last decade. Given V a suitable space of word-vectors, subspace of $\mathbb{R}^n$, then a space of useful predicates over V is given by $(\lambda x.\lambda z.z \cdot x > 0)\wr V \wr$.

Let us consider an example where V is a multivariate gaussian in $\mathbb{R}^n$ whose density is written G. The probability of a random word z satisfying a random predicate p is calculated as follows (because $\mathsf{measure}\Sigma(z : V)\Sigma(p : P) = 1$).

$$[\![\Sigma(z : V)\Sigma(p : P).\mathsf{IsTrue}(p(z))]\!](\lambda x.1) = [\![V]\!]\lambda z.[\![\Sigma(p : P)\mathsf{IsTrue}(p(z))]\!](\lambda x.1) =$$
$$\int_{\mathbb{R}^2} G(z)[\![V]\!]\lambda p.[\![\mathsf{IsTrue}(p(z))]\!](\lambda x.1)dz = \int_{\mathbb{R}^2} G(z)[\![V]\!]\lambda y.[\![\mathsf{IsTrue}(z \cdot y > 0)]\!](\lambda x.1)dydz =$$
$$\int_{\mathbb{R}^2}\int_{\mathbb{R}^2} G(z)G(y)(\text{if } z \cdot y > 0 \text{ then } 1 \text{ else } 0)dydz = 0.5$$

5 Related work

Instead of what we present here, several authors have proposed to use a fully fledged probabilistic programming language for natural language semantics. Such a language is sufficiently powerful to do so — in particular we can translate LMS to it. However, because it is free-form (at least as powerful as a turing machine), studying the properties of its programs is difficult. In contrast, many LMS formulas have useful properties. For example, a formula ϕ is provable in the first-order theory of real closed fields if and only if its encoding using the above quantifiers is true in LMS. Several other useful properties will be presented at the conference.

6 Conclusion

LMS is a concise system. While is sufficiently powerful to express a number of natural language phenomena, it is restricted so that it is decidable. These qualities of LMS is a reason to hope that LMS can play a role in the foundations of natural language semantics, where probability undoubtedly plays a crucial role.

References

1. Barwise, J., Cooper, R.: Generalised quantifiers and natural language. Linguistics and Philosophy **4**, 159–219 (1981)
2. Fox, C., Lappin, S.: Foundations of Intensional Semantics. Blackwell (2005)
3. Shan, C.c., Ramsey, N.: Exact bayesian inference by symbolic disintegration. In: Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages. pp. 130–144. POPL (2017)