Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, November 2015

Probabilistic Type Theory and Natural Language Semantics

Robin Cooper[1], Simon Dobnik[1], Shalom Lappin[1,2], and Staffan Larsson[1], [1] *University of Gothenburg,* [2] *King's College London*

Published by CSLI Publications

LiLT volume 10, issue (4) November 2015

Probabilistic Type Theory and Natural Language Semantics

Robin Cooper[1], Simon Dobnik[1], Shalom Lappin[1,2], and Staffan Larsson[1], [1] *University of Gothenburg,* [2] *King's College London*

Abstract

Type theory has played an important role in specifying the formal connection between syntactic structure and semantic interpretation within the history of formal semantics. In recent years rich type theories developed for the semantics of programming languages have become influential in the semantics of natural language. The use of probabilistic reasoning to model human learning and cognition has become an increasingly important part of cognitive science. In this paper we offer a probabilistic formulation of a rich type theory, Type Theory with Records (TTR), and we illustrate how this framework can be used to approach the problem of semantic learning. Our probabilistic version of TTR is intended to provide an interface between the cognitive process of classifying situations according to the types that they instantiate, and the compositional semantics of natural language.

LiLT Volume 10, Issue (4), November 2015.
Probabilistic Type Theory and Natural Language Semantics.
Copyright © 2015, CSLI Publications.

1 Probabilistic Semantics

In a classical semantic theory meaning is defined in terms of truth conditions. The meaning of a sentence is built up compositionally through a sequence of functions from the semantic values of constituent expressions to the value of the expression formed from these syntactic elements (Montague, 1974). Underspecified semantic theories weaken the connection between the meaning of a phrase and its constituents from a function to a relation (see Fox and Lappin, 2010 for discussion and references). Dynamic semantic accounts incorporate update procedures for incrementally computing the effect of new information on the interpretation of sentences in discourse and dialogue (see, for example, Kamp and Reyle, 1993). In all of these frameworks the type system is categorical. A type T identifies a set of possible denotations for expressions in T, and the system specifies combinatorial operations for deriving the denotation of an expression from the values of its constituents.

There are at least two problems with these frameworks. First, they cannot represent the gradience of semantic properties that is pervasive in speakers' judgements concerning truth, predication, and meaning relations. In general, predicates do not have determinate extensions (or intensions), and so, in many cases, speakers do not make categorical judgements about the interpretation of an expression. One might seek to attribute such gradience effects to performance factors, such as memory limitations and attentional focus, on the realization of semantic competence. Unless one can provide a precise explanation of performance mechanisms and how they produce the observed gradience phenomena, then such an account will have little if any explanatory content.

A second problem with the semantic theories that apply categorical type theories is that they offer no account of semantic learning. There is a fair amount of evidence indicating that language acquisition in general crucially relies on probabilistic learning (see, for example, Chater and Manning, 2006, Chater and Vitányi, 2007, Clark, 2007, Perfors et al., 2011, Clark and Lappin, 2011). It is not clear how a reasonable account of semantic learning could be constructed on the basis of the categorical type systems that either classical or revised semantic theories assume. Such systems do not appear to be efficiently learnable from the primary linguistic data (with weak learning biases), nor is there much psychological data to suggest that they are, themselves, biologically determined constraints on semantic learning.

A semantic theory that assigns probability rather than truth conditions to sentences is in a better position to deal with both of these issues. Gradience is intrinsic to the theory by virtue of the fact that speakers assign values to declarative sentences in the continuum of real numbers [0,1], rather than Boolean values in $\{0,1\}$. Moreover, a probabilistic account of semantic learning is facilitated if the target of learning is a probabilistic representation of meaning.

We consider two strategies for constructing a probabilistic semantics. One is a top-down approach where one sustains classical (categorical) type and model theories, and then specifies a function that assigns probability values to the possible worlds that the model provides. The probability value of a sentence relative to a model M is the sum of the probabilities of the worlds in which it is true. The other is a bottom-up approach where one defines a probabilistic type theory and characterizes the probability value of an Austinian proposition relative to a set of situation types (Cooper, 2005a). This proposition is the output of the function that applies to the probabilistic semantic type judgements associated with the syntactic constituents of the proposition.

1.1 Probability Distributions over Worlds

The top-down strategy is adopted by van Eijck and Lappin (2012). They retain a classical type theory and the specification of intensions for each type as functions from worlds to extensions. They define a *probabilistic model* M as a tuple $\langle D, W, P\rangle$ with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $P(w) \in [0,1]$, and $\sum_{w\in W} P(w) = 1$.[1]

An interpretation of a language L in an L-model $M = \langle D, W, P\rangle$ is given in terms of the standard notion $w \models \phi$, as follows:

$$[\![\phi]\!]^M \quad := \quad \sum\{P(w) \mid w \in W \wedge w \models \phi\}$$

This definition of a model entails that $[\![\neg\phi]\!]^M = 1 - [\![\phi]\!]^M$. Also, if $\phi \models \neg\psi$, i.e., if $W_\phi \cap W_\psi = \emptyset$, then $[\![\phi \vee \psi]\!]^M = \sum_{w\in W_{\phi\vee\psi}} P(w) = \sum_{w\in W_\phi} P(w) + \sum_{w\in W_\psi} P(w) = [\![\phi]\!]^M + [\![\psi]\!]^M$. These equations satisfy the axioms of Kolmogorov's (1950) probability calculus.

This theory has several attractive properties. It retains a classical type system and model theory to compute the value of a sentence in a world, and it is specified in such a way as to satisfy the axioms of a standard probability calculus for computing the probability of a sentence. Therefore, it uses well understood formal systems at both levels

[1] See(Gaifman and Snir, 1982) on the assignment of probabilities to worlds, understood as models.

of representation. It also proposes the outline of a theory of semantic learning for simple one-place predicate classifiers, where this could be generalised to a richer representation language.

However, it also suffers from the disadvantage that it requires probabilities to be assigned to entire worlds in the model. Complete worlds are not tractably representable. Assume that worlds are maximal consistent sets of propositions (Carnap, 1947).[2] If the logic of propositions is higher-order, then the problem of determining membership in such a set is not complete by virtue of the fact that the theorems of higher-order logic are not enumerable. But then it is not possible to identify all of the valid sentences, which hold across all worlds.

If the logic is classically first-order, then the membership problem is complete, but it is undecidable. It is possible, at least in principle, to list the set of valid sentences, which hold in all worlds, but not the set of invalid ones, which do not.

Alternatively, we could limit ourselves to propositional logic, which is both complete and decidable. We might try to generate a maximally consistent set of propositions by starting with a single finite proposition P that encodes a finite set of propositions that are true in a world w, where we formulate P in Conjunctive Normal Form (CNF). A formula in CNF is a conjunction of disjunctions of literals, and a literal is either an elementary proposition or the negation of one. We could then extend our coverage of the facts of w by simply adding true conjuncts to P. But it is not clear what (finite) set of rules or procedures we could use to decide which propositions to add in order to generate a full description of a world in a systematic way. Nor is it obvious at what point the conjunction will constitute a complete description of the world. By contrast, we can characterise a language L through a formal grammar that generates all and only the infinite set of sentences in L, from a finite set of rules. But with a world we do not have a comparable procedure for producing the infinite CNF required to provide a complete description of w.

Moreover, all the propositions that P entails must be added to it, and all the propositions with which P is inconsistent must be excluded, in order to obtain the maximal consistent set of propositions that describe a world. But then testing the satisfiability of P is an instance of the $kSAT$ problem, which, in the general case, is NP-complete.[3]

[2] Fox and Lappin (2005), Pollard (2008), generalizing on the characterizations of possible worlds in Carnap (1947), Jonsson and Tarski (1951), Kripke (1959), define worlds as ultrafilters in a prelattice of propositions, where the preorder specifies entailment among equivalence classes of propositions.

[3] The $kSAT$ problem is to determine whether a formula in propositional logic has

Why should the fact that constructing a description of a world is NP-complete be a problem? Isn't it sufficient that we can, in principle, encode all of the propositions that are true in w in a CNF formula, and exclude all the propositions that do not hold in w from this formula? Unfortunately this will not do if we are interested in sustaining the cognitive plausibility of our semantic theory. If this theory requires the representation of a set of worlds, and computing such a representation is not efficiently possible in real time, then we cannot regard this device as part of the mechanism through which humans construct interpretations of the sentences and other expressions in natural language.

Notice that the problem is not avoided by using a Kripke frame semantics (as in Montague, 1974) in which worlds are simple elements of a set W. In a system of this kind a model is an ordered k-tuple $\langle D, W, F, R\rangle$, where D is the domain of objects, F is an interpretation function that assigns intensions to the constants of a language, and R is an accessibility relation on W. Intensions are functions from worlds to denotations of the appropriate type. Propositions are functions from worlds to truth-values, and so every $w_i \in W$ is in a one-to-one correspondence with the maximal set $Prop_{w_i}$ of propositions that are true at w_i. But then each w_i is identified by its corresponding set of maximal propositions, and the problem of representing w_i reduces to that of determining membership in $Prop_{w_i}$.

In addition to the problems involved in representing individual worlds, enumerating the set of worlds raises difficulties of its own. The set of worlds is at least countably infinite. But Lewis (1973), as noted by Partee (1977), suggests that a reasonable cardinality for the set of possible worlds is transfinite: $\beth_2$. Rescher (1999) argues that possible worlds are inherently non-denumerable because they cannot be individuated. Given a specified world, there are an infinite number of variants that can be generated for that world by varying any of its property sets.

Some formal semanticists characterise intensions and modal concepts in terms of the set of possible situations rather than the set of possible worlds. Heim (1990), Lappin and Francez (1994), Lappin (2000), Kratzer (2014), among others, adopt this approach. As possible situations are parts of worlds it might appear, at first glance, that the representation problem which we encounter for worlds does not arise for possible situations.

In fact, positing the set of possible situations as the basis for an

a satisfying set of truth-value assignments. For the complexity results of different types of $kSAT$ problem see Papadimitriou (1995).

intensional semantics makes the representation problem significantly worse. As a world is a maximal consistent set of propositions, each situation in that world is a subset of this maximal set. But then each world w_i yields a power set $\mathcal{P}(Prop_{w_i})$ of possible situations. The maximal set of propositions that specify w_i is (at least countably) infinite, and the cardinality of $\mathcal{P}(Prop_{w_i})$ is higher than the cardinallity of $Prop_{w_i}$ (by Cantor's theorem for the cardinality of power sets). The set of possible situations is the union of $\mathcal{P}(Prop_{w_i})$ (more accurately, of $\mathcal{P}(Prop_{w_i}) - \emptyset$) for all w_i. This set is not recursively enumerable.

Computing the representation of an infinite set of worlds, or of possible situations, and constructing a probability distribution over them would seem to be an insuperable task. While assigning probabilities to elements of the set of worlds (or possible situations) may be formally viable, it is entirely unclear how we could estimate such a distribution computationally to derive interpretations of the sentences of a natural language.

The representation problem does not arise for individual situations, or for situation types (Barwise and Perry, 1983). Situations can be as large or as small as we need them to be. A situation type abstracts over situations through the use of parameters to represent possible elements of a situation.

1.2 Probability Distributions over Situation Types

Probability theorists working in AI often describe probability judgements as involving distributions over worlds. In fact, they tend to limit such judgements to a restricted set of outcomes or events, each of which corresponds to a partial world which is, effectively, a type of situation. A classic example of the reduction of worlds to situation types in probability theory is the estimation of the likelihood of heads vs tails in a series of coin tosses. Here the world is held constant except along the dimension of a binary choice between a particular set of possible outcomes. A slightly more complex case is the probability distribution for possible results of throwing a single die, which allows for six possibilities corresponding to each of its numbered faces.

Consider the following discussion from Halpern (2003).

> Most representations of uncertainty (certainly all the ones considered in this book) start with a set of *possible worlds*, sometimes called *states* or *elementary outcomes*. Intuitively, these are the worlds or outcomes that an agent considers possible. For example, when tossing a die, it seems reasonable to consider six possible worlds, one for each of the ways that the die could land. This could be represented by a set W consisting of six possible worlds, $\{w_1, ..., w_6\}$; the world w_i is the one

> where the die lands i, for i = 1,...,6. (The set W is often called *the sample space* in probability texts.)
>
> For the purposes of this book, the objects that are known (or considered likely or possible or probable) are *events* (or *propositions*). Formally, an event or proposition is just a set of possible worlds. (p. 12)

Clearly what Halpern is describing as a possible world here corresponds to a type of situation in the sense of situation theory (Barwise and Perry, 1983), rather than to the maximal worlds of possible worlds semantics. If we model worlds of the latter type algebraically, they are ultrafilters of propositions in a lattice of entailment. Each proposition in the language receives a truth-value in such a world. But in practice probability distributions are estimated over fragments of worlds specified by a highly constrained set of possible outcomes. The more fragments of worlds are observed, the better the estimation of the underlying hidden probability distribution that describes all the worlds.

We are making explicit the assumption, common to most probability theories used in AI, with clearly defined sample spaces, that probability is distributed over situation types, rather than over sets of worlds, understood as maximal consistent sets of propositions, or over the set of all possible situations.[4]

An Austinian proposition is a judgement that a situation is of a particular type, and we treat it as probabilistic. In fact, it expresses a subjective probability in that it encodes the belief of an agent concerning the likelihood that a situation is of that type. The core of an Austinian proposition is a type judgement of the form $s : T$, which states that a situation s is of type T. On our account this judgement is expressed probabilistically as $p(s : T) = r$, where $r \in [0,1]$.

In the probabilistic type system that we are proposing situation types are intensional objects over which probability distributions are specified. This allows one to reason about the likelihood of alternative states of affairs without invoking possible worlds or possible situations.

[4] (Goodman and Lassiter, 2015) also propose a probabilistic account of semantics, which represents the role of pragmatic factors in determining meaning in a given context. They treat the interpretation of expressions in a language as a process of reasoning under uncertainty, and they model this process in Bayesian terms. They describe a stochastic λ-calculus and indicate how it is implemented in the programming language Church. They show how Church functions can be used to assign probabilities to situation types, which they call possible worlds, and these functions provide the basis for their formalisation of the meanings of predicates.

Goodman and Lassiter use a classical type theory of the sort applied in Montague semantics to the meanings of syntactically complex expressions. By contrast, we are proposing a rich probabilistic type theory as the mechanism for compositional semantic interpretation.

Our theory assumes only actual situations, and an intensional type system. Types are not sets of situations. They can be as large and underspecified, or as small and fine-grained as we require them to be. It is not necessary to represent the full set of situations (actual or possible) in order to acquire these types. They are classifiers of situations that can be learned through sampling of actual situations, and probabilistic reasoning concerning the types to which they belong. Therefore, the problems of tractable representation that we encountered with worlds, and with the set of possible situations, do not arise in the semantic theory that we develop here.

The simplest probability judgements involve the application of a binary classifier to a situation (state or event) to estimate the likelihood that it is of type T or of type T', the complement of T. These two types exhaust the probability space, so that $p(s : T) + p(s : T') = 1$. Estimating the result of a coin toss is an instance of this sort of binary classifier. In more complex cases, judgements distribute probability among k alternative situation types, as in the outcome of a die throw. Some of the complexity involved in estimating probability consists in determining the range of alternative situations types over which probability is distributed, i.e. identifying the mutually exclusive and exhaustive types, $T_1, \ldots, T_k$, that form a category (a hypothesis space) such that $p(s : T_1) + ... + p(s : T_k) = 1$.

Probability judgements are often conditioned by other judgements. One estimates the likelihood that a situation s is of type T_1, given that it is of type T_2, where a conditional probability judgement of this kind is expressed as $p(s : T_1 \mid s : T_2)$.

Assume that A and B are events that may not be independent of each other. In classical probability theory the following equation defines the probability of A and B co-occurring, in terms of the conditional probability of A relative to B.

$$p(A \wedge B) = p(A)p(B \mid A)$$

This equation entails the following equation for conditional probability.[5]

$$p(A \mid B) = \frac{p(A \wedge B)}{p(B)}$$

Formulating this equation for conditional probability in terms of judgements concerning situation types we obtain

$$p(s : T_1 \mid s : T_2) = \frac{p(s : T_1 \wedge T_2)}{p(s : T_2)}$$

[5] See, for example, Halpern (2003), Section 3.2 for a discussion of conditional probability.

A common procedure for estimating $p(A \mid B)$ is Maximum Likelihood Estimation (MLE), which divides the number of co-occurrences of A and B events by the number of occurrences of B events in a sample of observations

$$\frac{|A \wedge B|}{|B|}$$

For $p(s^* : T_1 \mid s^* : T_2)$, where s^* is a particular situation, this would involve determining, for a set of observations,

$$\frac{|\{s \mid s : T_1 \wedge T_2\}|}{|\{s \mid s : T_2\}|}.$$

In Section 4 we suggest a modified version of MLE to compute $p(s^* : T_1 \mid s^* : T_2)$.

If a simple binary classifier is conditioned, then we are estimating $p(s^* : T_1 \mid s^* : T_2)$ and $p(s^* : T_1' \mid s^* : T_2)$. For a conditioned k-ary classifier we are judging $p(s^* : T_1 \mid s^* : T)$ ... $p(s^* : T_k \mid s^* : T)$. In both cases the conditioning judgement can be represented as the mother node in a tree, and the conditioned judgements as its daughters, as in Bayesian networks of belief.[6] Such networks (which are directed acyclic graphs) encode systems of probabilistic belief update and reasoning in which information and inferences are passed in both directions along the paths of the graph. In future work we will explore the connection between our proposed type system and Bayesian networks. In this paper we focus on elementary Bayesian classifiers (which can be modelled as two-layer Bayesian networks) to illustrate how our type theory serves as an interface between perceptual judgement and semantic interpretation.

The bottom-up view of our probabilistic type theory avoids the representation problem of the top-down distribution over worlds approach by assigning probabilities to individual type judgements as classifier applications. The probability of a sentence (more accurately, the type judgement that it expresses) is, then, determined directly by the probabilities of its constituent types. We will specify a bottom-up probabilistic semantics on the basis of the type theory that we propose in Section 2.

This approach to probabilistic semantics is aligned with the way probabilities have been used in language technology. For example, a statistical language model is a probability distribution over a sequences of words. Importantly, it is a sampling distribution of the underlying distribution that it is attempting to approximate. The top down approach described in the previous section would correspond to the underlying distribution. The reason why language technology uses sample distri-

[6] See Pearl (1990) and Halpern (2003) on Bayesian networks.

butions is that the underlying distribution of a language can never be observed. The number of the utterances that speakers of a language can make is infinite. But even if the number of utterances of a language were finite, it is unclear whether an agent using such a language would be able to observe all of the utterances in his/her lifetime to build a global probabilistic model (Fagin et al., 1995). Furthermore, changes are introduced to language as new objects and events come into existence and agents interact with each other. This means that global probability distributions over words would have to be updated at each such temporal instance when changes are introduced.

Therefore, in practical applications (which include language technology) we must deal with sample probability distributions which are partial and incrementally updatable through learning. Inference for knowledge is provided by Bayesian reasoning. An agent not being able to build such updatable sample distributions would not be able to deal with any real-world situations. An example of this comes from robotic localisation map building, where such models are widely used (Dissanayake et al., 2001). Furthermore, there have been proposals in cognitive science (Tenenbaum et al., 2011) that the human mind also works in the same way. The approach taken in this paper to probabilistic semantics is similar in spirit to both of these lines of work. (Dobnik et al., 2013) explores a connection between robotic localisation and type theoretic semantics.

By relating our theoretical framework for probabilistic semantics with the way in which probabilistic language modelling is used in language technology we allow for the application of insights from linguistics to achieve improved probabilistic models of natural language. On one hand the types allow us to encode and compose conceptual knowledge. On the other, we can employ them to capture how such knowledge is learned in a data-driven way, through probabilistic modelling, both from the representation of words in linguistic interaction, and from the application of words to sensory observations. Previous attempts to relate language and perception through some form of semantic representation often make use of a variant of first-order logic (see for example the work of Matuszek et al. (2012b) for mapping natural language to a robotic control language, and Matuszek et al. (2012a) for the relation between logic and perception).

Our approach diverges from this work in using probabilistic types and type judgements as the primary elements semantic representation, rather than first-order formulas with models. Our probabilistic type system permits us to construct a unified framework for expressing probabilistic reasoning from sensory projections to compositional sentence meanings, and to representations of dialogue (Dobnik et al., 2014). In this paper we describe this framework. The type theory provides the foundation for an account of semantic composition, which we present in Section 3. It also grounds our sketch of semantic learning, where individual classifiers are acquired probabilistically through observation driven Bayesian inference and update rules. We outline this view of learning in Section 4.

2 Rich Type Theory as a Theory of Probabilities

By *rich type theory* we understand something analogous to what Luo (2010, 2011) calls “modern type theory”. Whereas the type theory that Montague (1974) used for natural language semantics was based on Church’s (1940) simple theory of types, containing types for entities, truth values and all possible functions which can be constructed from these, rich type theories such as Martin-Löf’s (1984) intuitionistic type theory, embrace the idea that any intuitive “proposition” should be represented by a type. In linguistic terms we can think of a type corresponding to a “proposition” as a type of situation or event (as suggested by Ranta, 1994). Thus the semantic content of *Kim saw a dog* will be the type of situations where Kim saw a dog. Rich type theories are standardly formulated as theories of categorical type judgements. An object a either is or is not of a type T. However, it seems to us that such type theories provide the basis for a theory of probabilistic type judgements. There is a probability p that an object a is of type T, or that there is some object of type T. Alternatively, and perhaps more interestingly for us, there is a probability p that an agent A makes a judgement that an object a is of type T, or that there is some object of type T.

We will formulate our proposal in terms of a particular adaptation of rich type theory, Type Theory with Records (TTR, Cooper, 2005a, 2012). We have used this previously for natural language semantics (see, for example, Cooper, 2005b, 2012), and to analyze semantic coordination and learning (for example, Larsson and Cooper, 2009, Cooper and Larsson, 2009). It has also been applied to the analysis of interaction in dialogue (for example, Ginzburg, 2012), and in modelling robotic states and spatial cognition (for example, Dobnik et al., 2013). We believe that a probabilistic version of TTR could be useful in all these domains.

Central to standard formulations of rich type theories is the notion of a judgement $a : T$, that object a is of type T. We will represent the probability of this judgement as $p(a : T)$. We will now show how various kinds of types in TTR can be associated with probabilities.

Basic Types. TTR introduces basic types, that is types which are not constructed out of other objects introduced in the theory. If T is a basic type, we assume for the moment that $p(a : T)$ for any object a is provided by a probability model, an assignment of probabilities to judgements involving basic types. We will discuss what such models might be in Section 4.

PTypes. TTR has types, called *ptypes*, which are constructed from a *predicate* and an appropriate sequence of arguments. The appropriateness of a sequence of arguments for a predicate is determined by the *arity* associated with the predicate. The arity is a sequence of types (or, if we wish to allow polymorphic predicates, a set of sequences of types) to which the sequence of arguments must belong. An example might be the predicate 'man' with arity $\langle \mathit{Ind}, \mathit{Time} \rangle$, where the types *Ind* and *Time* are the type of individuals and of time points respectively. Thus man(a,t) would be the type of situation (or eventuality) where a is a man at time t. We will assume for now that a probability model will provide probabilities $p(e : r(a_1, \ldots, a_n))$ for ptypes $r(a_1, \ldots, a_n)$, and we will return to this issue in Section 4.

There are two kinds of probabilistic type judgements in TTR. $p(a : T)$ is the probability that a is of type T, while $p(T)$ is the probability that T is not empty. For any type T it should be the case that $p(a : T) \leq p(T)$. $p(T)$ is the disjunctive probability of $p(a_i : T)$ for all a_i, or at least an estimation of this in the limit where there are infinitely many a_i's.[7]

Meets and Joins. For any types T_1 and T_2 TTR says that there exists the meet, $T_1 \wedge T_2$ and the join $T_1 \vee T_2$. $a : T_1 \wedge T_2$ just in case $a : T_1$ and $a : T_2$. $a : T_1 \vee T_2$ just in case either $a : T_1$ or $a : T_2$ (possibly both).[8]

Note that TTR is an intensional type theory. The meet and join operations create many distinct types which are equivalent. So, for example, for two different types T_1 and T_2, $T_1 \wedge T_2$ and $T_2 \wedge T_1$ are distinct types, although they are equivalent in the sense that no matter what we assign to the basic types, for any object a, $a : T_1 \wedge T_2$ if and only if $a : T_2 \wedge T_1$. Similarly for any type T, $T \wedge T$ will be a type distinct from T, although the two types are equivalent. Similar remarks hold

[7] See our discussion of disjunctive probability and probabilistic disjunctive types below for a characterization of the estimating such a probability in the limit.

[8] This use of intersection and union types is not standard in rich type theories, where product and disjoint union are preferred following the Curry-Howard correspondence for conjunction and disjunction.

for join types.

The probabilities for meet and joint types are defined by the classical Kolmogorov (1950) equations for conjunction and disjunction.

$p(a : T_1 \wedge T_2) = p(a : T_1)p(a : T_2 \mid a : T_1)$[9]

$p(a : T_1 \vee T_2) = p(a : T_1) + p(a : T_2) - p(a : T_1 \wedge T_2)$

These definitions sustain the equivalences:

$p(a : T_1 \wedge T_2) = p(a : T_2 \wedge T_1)$

$p(a : T_1 \vee T_2) = p(a : T_2 \vee T_1)$

$p(a : T \wedge T) = p(a : T) = p(a : T \vee T)$

We need to consider the fact that the types in a meet or join may stand in the subtype relation. A type T_1 is a subtype of type T_2, $T_1 \sqsubseteq T_2$, just in case $a : T_1$ implies $a : T_2$ no matter what we assign to the basic types. If $T_1 \sqsubseteq T_2$ then $a : T_1 \wedge T_2$ iff $a : T_1$, and $a : T_1 \vee T_2$ iff $a : T_2$. Similarly, if $T_2 \sqsubseteq T_1$ then $a : T_1 \wedge T_2$ iff $a : T_2$, and $a : T_1 \vee T_2$ iff $a : T_1$.

It is reasonable to assume that if $T_1 \sqsubseteq T_2$, then $p(a : T_2 \mid a : T_1) = 1$. Given this assumption, our definitions of meet and join types support the following:

If $T_1 \sqsubseteq T_2$, then $p(a : T_1 \wedge T_2) = p(a : T_1)$, and $p(a : T_1 \vee T_2) = p(a : T_2)$.

If $T_2 \sqsubseteq T_1$, then $p(a : T_1 \wedge T_2) = p(a : T_2)$, and $p(a : T_1 \vee T_2) = p(a : T_1)$.

If $T_1 \sqsubseteq T_2$, then $p(a : T_1) \leq p(a : T_2)$.

These definitions also entail:

$p(a : T_1 \wedge T_2) \leq p(a : T_1)$

$p(a : T_1) \leq p(a : T_1 \vee T_2)$

Conjunctive and Disjunctive Probabilities We generalize probabilistic meet and join types to probabilities for unbounded conjunctive and disjunctive type judgements, again using the classical Kolmogorov equations.

Let $\bigwedge_p(\{a_0 : T_0, \ldots, a_n : T_n\})$ be *the conjunctive probability of judgements* $a_0 : T_0, \ldots, a_n : T_n$.

$\bigwedge_p(\{a_0 : T_0, \ldots, a_n : T_n\}) =$

$\bigwedge_p(\{a_0 : T_0, \ldots, a_{n-1} : T_{n-1}\})p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1})$, if

[9] If T_1 and T_2 are probabilistically independent, we have $p(a : T_1 \wedge T_2) = p(a : T_1)p(a : T_2)$.

$n > 0$,
$p(a_0 : T_0)$, if $n = 0$

$$\bigwedge_p(\{\}) = 1$$

We interpret universal quantification as an unbounded conjunctive probability, which is true if it is vacuously satisfied ($n = 0$).[10]

The conjunctive probability of an infinite sequence of judgements is $\bigwedge_p(\{a_0 : T_0, a_1 : T_1, \ldots\}) = \lim_{i \to \infty} \bigwedge_p(a_i : T_i)$.

Let $\bigvee^p(\{a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n\})$ be the disjunctive probability of judgements $a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n$. It is computed by

$\bigvee^p(\{a_0 : T_0, \ldots, a_n : T_n\}) =$

$\bigvee^p(\{a_0 : T_0, \ldots, a_{n-1} : T_{n-1}\}) + p(a_n : T_n) - \bigwedge_p(\{a_0 : T_0, \ldots, a_{n-1} : T_{n-1}\})p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1})$, if $n > 0$,
$p(a_0 : T_0)$, if $n = 0$

$$\bigvee^p(\{\}) = 0$$

We take existential quantification to be an unbounded disjunctive probability, which is false if it lacks a single non-nil probability instance.

The disjunctive probability of an infinite sequence of judgements is $\bigvee^p(\{a_0 : T_0, a_1 : T_1, \ldots\}) = \lim_{i \to \infty} \bigvee^p(a_i : T_i)$.

Conditional Conjunctive Probabilities. We compute conditional conjunctive probabilities with the following equation.

$\bigwedge_p(\{a_0 : T_0, \ldots, a_n : T_n\} \mid a : T) =$

$\bigwedge_p(\{a_0 : T_0, \ldots, a_{n-1} : T_{n-1}\} \mid a : T)p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1}, a : T))$, if $n > 0$,
$p(a_0 : T_0 \mid a : T)$, if $n = 0$

$$\bigwedge_p(\{\} \mid a : T) = 1$$

[10] See Paris (2010) on computing the probability of a universally quantified statement as the product of the probabilities of its corresponding individual instances in a conjunction.

Function Types. For any types T_1 and T_2 TTR gives us a type $(T_1 \rightarrow T_2)$, which is the type of total functions with domain the set of all objects of type T_1 and range included in objects of type T_2.

We require that there is at least one function for each function graph, provided that it is well-typed, such that its domain (left projection) is the set of objects belonging to some type and its range (right projection) is a subset of the set of objects belonging to some type.[11] This will help us associate probabilities with judgements concerning function types. The following formula says that the probability that a function f is of type $(T_1 \rightarrow T_2)$ is the probability that everything in its domain is of type T_1 and that everything in its range is of type T_2, and furthermore that everything not in its domain which has some probability of being of type T_1 is *not* in fact of type T_1.

We can compute the probability that a function belongs to a certain function type as follows.

$$p(f : (T_1 \rightarrow T_2)) =$$
$$\underset{a \in \mathrm{dom}(f)}{\bigwedge_{p}} (a : T_1, f(a) : T_2)(1 - \underset{p(a:T_1)>0,\ a \notin \mathrm{dom}(f)}{\bigvee^{p}} (a : T_1))$$

Suppose that T_1 is the type of event where there is a flash of lightning and T_2 is the type of event where there is a clap of thunder. Suppose that f maps lightning events to thunder events, and that it has as its domain all events which have been judged to have probability greater than 0 of being lightning events. Let us consider that all the putative lightning events were clear examples of lightning (i.e. judged with probability 1 to be of type T_1) and are furthermore associated by f with clear events of thunder (i.e. judged with probability 1 to be of type T_2). Suppose there were four such pairs of events. Then the probability of f being of type $(T_1 \rightarrow T_2)$ is $(1 \times 1)^4$, that is, 1.

Suppose, alternatively, that for one of the four events f associates the lightning event with a silent event, that is, one whose probability of being of T_2 is 0. Then the probability of f being of type $(T_1 \rightarrow T_2)$ is $(1 \times 1)^3 \times (1 \times 0) = 0$. One clear counterexample is sufficient to show that the function is definitely not of the type.

In cases where the probabilities of the antecedent and the consequent type judgements are higher than 0, the probability of the entire judgement on the existence of a functional type f will decline in proportion to the size of dom(f). Assume, for example that there are k elements $a \in \mathrm{dom}(f)$, where for each such a $p(a : T_1) = p(f(a) : T_2) \geq .5$. Every

[11] Since the complete version of TTR allows both singleton types and join types it will be possible to construct a function type for any set of ordered pairs, provided that the objects in the pairs belong to some type.

a_i that is added to dom(f) will reduce the value of $p(f : (T_1 \rightarrow T_2))$, even if it yields higher values for $p(a : T_1)$ and $p(f(a) : T_2)$. This is due to the fact that we are treating the probability of $p(f : (T_1 \rightarrow T_2))$ as the likelihood of there being a function that is satisfied by all objects in its domain. The larger the domain, the less probable that all elements in it fulfill the functional relation.

We are, then, interpreting a functional type judgement of this kind as a universally quantified assertion over the pairing of objects in dom(f) and range(f). The probability of such an assertion is given by the conjunction of assertions corresponding to the co-occurrence of each element a in f's domain as an instance of T_1 with $f(a)$ as an instance of T_2. This probability is the product of the probabilities of these individual assertions.

This seems reasonable, but it only deals with functions whose domain is all objects which have been judged to have some probability, however low, of being of type T_1. Intuitively, functions which leave out some of the objects with lower likelihood of being of type T_1 should also have a probability of being of type $(T_1 \rightarrow T_2)$. This factor in the probability is represented by the second element of the product in the formula.

Function Argument Application. There are additional probabilities associated with a function. Given that $f : (T_1 \rightarrow T_2)$ and $a : T_1$, what is the probability that $f(a) : T_2$? This should be $p(f(a) : T_2 \mid a : T_1, f : (T_1 \rightarrow T_2))$. Probabilities associated with functions play an important role in probabilistic reasoning. Suppose that e is an event where there is a loud bang and T_2 is the type of events where there is thunder. $p(e : T_2)$ may be quite low considered independently. There are many loud bangs that are not thunder. However, suppose that T_1 is the type of lightning events and that event a is a lightning event, $a : T_1$. Suppose furthermore that $f(a) = e$ and the probability that $f : (T_1 \rightarrow T_2)$ is high. Then $p(f(a) : T_2 \mid a : T_1, f : (T_1 \rightarrow T_2))$ should be high. Functional types provide us with a way of generating conditional probabilities that correspond to defeasible inferences.

Negation. The negation, $\neg T$, of type T, is the function type $(T \rightarrow \bot)$ where $\bot$ is a necessarily empty type (that is, a type which has no witnesses regardless of what we assign to basic types). $p(\bot) = 0$.

It is important to make clear the intuition behind this negation which is derived from the standard approach to negation in intuitionistic logic. It is a negation that says that there is no witness for the type T. For example, if T is the type of situation where Kim is asleep, $\neg T$ is a type that would have a witness just in case there are no situations in which Kim is asleep, that is, T is empty. There are alternative intuitions which could be called "negation" in a type system, for example, where an object of type $\neg T$ is of a type incompatible with T (Cooper and Ginzburg, 2011, 2012).

It follows from our rules for function types that

$p(f : \neg T) = 1$ if $\mathrm{dom}(f) = \emptyset$, that is T is empty, and 0 otherwise.

If $\neg T$ is non-empty, then there must be a function on the set of objects of type T whose range is included in the set of objects of type $\perp$, that is the empty set. The only way that a function can have the empty set as its range is if its domain is also the empty set. Therefore, if $\neg T$ is non-empty then T must be empty.

Dependent Types. In TTR dependent types are functions from objects to types. Given appropriate arguments as functions they will return a type. Therefore, the discussion of probabilities associated with functions above under **Function Types** and **Function Argument Application** also applies to dependent types.

Record Types. A record in a type system associated with a set of labels is a set of ordered pairs (*fields*) whose first member is a label and whose second member is an object of some type (possibly a record type). Records are required to be functional on labels, that is each label in a record can only occur once in the record's left projection. If r is a record containing a field with the label ℓ, then we use $r.\ell$ to denote the object contained in that field in r. Since the objects in the fields may themselves be records we may have a sequence of labels pointing to an object in a record embedded in r and we will extend the notation to allow expressions of the form $r.\ell_1.\ell_2. \ldots .\ell_n$. We will refer to $\ell_1.\ell_2. \ldots .\ell_n$ as a *path* in r.

A dependent record type is a set of fields (ordered pairs) consisting of a label ℓ followed either (clause 2) by a type or (clause 3) by a pair consisting of a dependent type (a function of one or more arguments which returns a type) and a sequence of paths (indicating where in a record the arguments to the dependent type are to be found). The set of record types is defined by:

1. $[]$, that is the empty set or *Rec*, is a record type. r : *Rec* just in case r is a record.

2. If T_1 is a record type, ℓ is a label not occurring in T_1, and T_2 is a type, then $T_1 \cup \{\langle \ell, T_2 \rangle\}$ is a record type. $r : T_1 \cup \{\langle \ell, T_2 \rangle\}$ just in case $r : T_1$, $r.\ell$ is defined (ℓ occurs as a label in r) and $r.\ell : T_2$.

3. If T is a record type, ℓ is a label not occurring in T, $\mathcal{T}$ is a dependent type requiring n arguments, and $\langle \pi_1, \ldots, \pi_n \rangle$ is an n-

place sequence of paths in T,[12] then $T \cup \{\langle \ell, \langle \mathcal{T}, \langle \pi_1, \ldots, \pi_n\rangle\rangle\rangle\}$ is a record type. $r : T \cup \{\langle \ell, \langle \mathcal{T}, \langle \pi_1, \ldots, \pi_n\rangle\rangle\rangle\}$ just in case $r : T$, $r.\ell$ is defined and $r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n)$.

We introduce a type, *RecType*, the type of record types, such that T:*RecType* just in case T is a record type as defined above.

We define the probability that an object r is of a record type T with these clauses:

1. $p(r : \mathit{Rec}) = 1$ if r is a record, 0 otherwise
2. $p(r : T_1 \cup \{\langle \ell, T_2\rangle\}) = \bigwedge_p (r : T_1, r.\ell : T_2)$ if $r.\ell$ is defined, 0 otherwise
3. If $\mathcal{T} : (T_1 \to (\ldots \to (T_n \to T')\ldots))$,
   then for any record type, T, $p(r : T \cup \{\langle \ell, \langle \mathcal{T}, \langle \pi_1, \ldots, \pi_n\rangle\rangle\rangle\}) =$
   $$\bigwedge_p (r : T, r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n) \mid r.\pi_1 : T_1, \ldots, r.\pi_n : T_n)$$

We use a tabular format to represent records and record types. A record $\{\langle \ell_1, v_1\rangle, \ldots, \langle \ell_n, v_n\rangle\}$ is displayed as

$$\begin{bmatrix} \ell_1 & = & v_1 \\ \ldots & & \\ \ell_n & = & v_n \end{bmatrix}$$

and a record type $\{\langle \ell_1, T_1\rangle, \ldots, \langle \ell_n, T_n\rangle\}$ is displayed as

$$\begin{bmatrix} \ell_1 & : & T_1 \\ \ldots & & \\ \ell_n & : & T_n \end{bmatrix}$$

Probabilities that Types are Non-Empty. We also consider judgements that a type T is non-empty, that is, that there is some a such that $a : T$. This judgement is sometimes represented in type theory as 'T true'. When types are considered as "propositions", then they are true just in case there is something of the type, and false otherwise. For example, *Kim saw a dog* is true just in case the type of situation where Kim saw a dog is non-empty. We represent the probability of a type T being non-empty as $p(T)$. For any type T it is the case that $p(a : T) \leq p(T)$.

We define $p(T)$ to be $\overset{\infty}{\underset{i}{\overset{p}{\bigvee}}}(a_i : T)$ (that is, $\lim_{i \to \infty} \overset{p}{\bigvee}(a_i : T)$) for a_i such that $p(a_i : T) > 0$. It does not matter which sequence of a's you choose,

[12] In the full version of TTR we also allow absolute paths which point to particular records, but we will not include them here.

as the disjunction of all the probability sequences will converge on the same value.

Remark: $p(\neg T) = 1 - p(T)$

Proof By definition of $p(\neg T)$,

$$\begin{aligned} p(\neg T) &= p(T \to \bot) \\ &= \overset{\infty}{\underset{i}{{\bigvee}^{p}}}(f_i : T \to \bot) \text{ for } p(f_i : T \to \bot) > 0. \end{aligned}$$

For each f_i,

$$p(f_i : T \to \bot) = \underset{a\in\mathrm{dom}(f_i)}{{\bigwedge}_{p}}(a : T, f_i(a) : \bot)(1 - \underset{a\notin\mathrm{dom}(f_i)}{{\bigvee}^{p}}(a : T))$$

Since for any a, $p(a : \bot) = 0$, the only cases where $p(f_i : T \to \bot) > 0$ will be those in which $\mathrm{dom}(f_i) = \emptyset$. Therefore

$$\begin{aligned} \underset{a\in\mathrm{dom}(f_i)}{{\bigwedge}_{p}}(a : T, f_i(a) : \bot) &= {\bigwedge}_{p}\{\} \\ &= 1 \end{aligned}$$

This means that for functions $f^{\emptyset}$ with empty domains

$$p(f^{\emptyset} : T \to \bot) = 1 - \underset{a}{{\bigvee}^{p}}(a : T)$$

which is the probability that nothing is of type T. Given that

$$\underset{a}{{\bigvee}^{p}}(a : T) = \overset{\infty}{\underset{i}{{\bigvee}^{p}}}(a_i : T) = p(T)$$

we have shown that

$$p(f^{\emptyset} : T \to \bot) = 1 - p(T)$$

As we are assuming that there is at least one function for each function graph, $p(f^{\emptyset} : T \to \bot)$ for a particular function, $f^{\emptyset}$, will be identical with $p(T \to \bot)$. The argument for this is as follows: assuming we have found one such function, $f^{\emptyset}$, any other candidate functions will not change the probability. There are two cases to consider. (i) If the domain of the new function is empty, then it will have the same probability as the first function of being of type $(T \to \bot)$. The conditional probability of it being of this type given that the first function is of this type is 1. This means that the disjunctive probability will not change. (ii) The new function has a non-empty domain. But this has a 0 possibility (both absolute and conditional) of being of a function type with $\bot$ as the range. Hence such a function will not be considered among those which have greater than 0 probability of being a witness for the function type in question.

Thus we have shown $p(\neg T) = 1 - p(T)$, as desired. □

Corollary: (i) $p(T \vee \neg T) = 1$, and (ii) $p(\neg\neg T) = p(T)$.

Proof (i) follows from $p(T) = \bigvee\limits_{i}^{\infty}{}^{p}(a_i : T)$, our equations for the probability of disjunctive type judgements, and the Remark.
$p(\neg T) = 1 - p(T)$, and $p(T \vee \neg T) = p(T) + p(\neg T) = 1$.
(ii) By the Remark,
$p(\neg\neg T) = 1 - p(\neg T)$, and $1 - p(\neg T) = 1 - (1 - p(T)) = p(T)$. □

The classical Kolmogorov equations that we use to specify probability judgements for negative and disjunctive types in our system preserve Boolean negation.

In the case of dependent types there is an additional probability judgement that we can consider: what is the likelihood that there is something of the type resulting from the application of the dependent type to an argument? Suppose that $\mathcal{T}$ is the dependent type of thunder events caused by a particular lightning event (an event of type T_1). Then the relevant probability is $p(\mathcal{T}(a) \mid a : T_1)$. Some dependent types are more reliable than others. In the case of thunder and lightning the probability of there being a thunder event associated with a particular lightning event is, let us say, 1 (it depends on whether you are including inaudible thunder events). In such a case, given a flash of light, the probability that a clap of thunder will follow is the same as the probability that the flash of light was lightning. Suppose in contrast that T_1 is the type of actors and $\mathcal{T}(a)$ is the type of situation where a wins an Oscar. Here $p(\mathcal{T}(a) \mid a : T_1)$ is much lower than 1, and it is (perhaps) computed on the basis of the percentage of actors who win Oscars. The probability of an arbitrary person winning an Oscar is increased if they are an actor, but being an actor by no means guarantees you an Oscar. Thus an important part of learning to reason effectively has to do with computing the conditional probabilities associated with dependent types. We address this issue in Section 4.

3 Semantic Composition

3.1 Interpretation with Categorical Types

We will illustrate our approach by defining a toy fragment. We generate the fragment with standard context-free phrase structure rules, and then we define an interpretation function $[\![\cdot]\!]$ for this fragment.[13] In

[13] This differs from the standard TTR approach to grammar, which uses signs and includes syntax in the realm of TTR. Here we concentrate on the TTR treatment

this subsection, we illustrate how TTR provides the basis for a compositional semantics by specifying the interpretation of the phrases and sentences of the fragment in classical terms. Semantic rules apply functions to type theoretically defined arguments to return categorical interpretations, which, in the case of a sentence S, is intuitively a type of situations. In Section 3.2 we modify this system so that semantic rules apply to type theoretic arguments to give values defined in terms of probabilistic type judgements.

We characterize our toy fragment in terms of the following context-free grammar.

$$\text{S} \rightarrow \text{S} \left\{ \begin{array}{c} \text{and} \\ \text{or} \end{array} \right\} \text{S}$$

S → Neg S
S → NP VP
NP → Det N
NP → Det N'
NP → N_{prop}
N' → ADJ N'
N' → N
VP → V_t NP
VP → V_i

Neg → “it’s not true that”
Det → “a” | “some” | “every” | "most"
N → “boy” | “girl”
ADJ → "green" | "imaginary"
N_{prop} → “Kim” | “Sandy”
V_t → “knows” | “sees”
V_i → “smiles” | “laughs”

$\mathit{Ppty} = (\left[\text{x}{:}\mathit{Ind}\right]{\rightarrow}\mathit{RecType})$ – the type of properties of individuals.
$\mathit{Quant} = (\mathit{Ppty}{\rightarrow}\mathit{RecType})$ – the type of quantifiers.

kim, sandy : *Ind.*
know, see are predicates with arity ⟨*Ind*, *Ind*⟩.
boy, girl, smile, laugh are predicates with arity ⟨*Ind*⟩.
some, every, most are predicates with arity ⟨*Ppty*, *Ppty*⟩.
green, imaginary are predicates with arity ⟨*Ind*, *Ppty*⟩.

Note that we follow Montague’s uniform intensional interpretation of adjectives (and other modifiers). The relational properties that modifiers express are not extensional on the properties to which they apply. But we can avoid meaning postulates in identifying the subclass of intersective modifiers. We can characterize the elements of this subclass

of interpretation and the modification of this account to accommodate probabilistic type judgements.

as functions that return a subtype of the argument to which they apply. So the interpretation of a *green car* will be based on a subtype of *car*, while that of *imaginary car* will not.

According to the definition of a system of types (Cooper, 2012) based on these predicates the set of ptypes will be:

$\{r(a) \mid r \in \{\text{boy,girl,smile,laugh}\} \wedge a : \mathit{Ind}\}$
$\cup\{r(a,b) \mid r \in \{\text{know,see}\} \wedge a, b : \mathit{Ind}\}$
$\cup\{q(Q,P) \mid q \in \{\text{some,every,most}\} \wedge Q, P : \mathit{Ppty}\}$
$\cup\{r(a,P) \mid r \in \{\text{green,imaginary}\} \wedge a : \mathit{Ind} \wedge P : \mathit{Ppty}\}$

$\llbracket\ [_{\text{S}}\ \text{S}_1 \text{ and } \text{S}_2]\ \rrbracket = \begin{bmatrix}\text{e}_1{:}\llbracket\ \text{S}_1\ \rrbracket \\ \text{e}_2{:}\llbracket\ \text{S}_2\ \rrbracket\end{bmatrix}$

(that is, a record type with two fields labelled by ‘e$_1$’ and ‘e$_2$’; a record of this type models a situation which has two component situations labelled by ‘e$_1$’ and ‘e$_2$’ which are of the types given by $\llbracket$ S$_1$ $\rrbracket$ and $\llbracket$ S$_2$ $\rrbracket$ respectively)

$\llbracket\ [_{\text{S}}\ \text{S}_1 \text{ or } \text{S}_2]\ \rrbracket = \begin{bmatrix}\text{e}{:}\llbracket\ \text{S}_1\ \rrbracket \vee \llbracket\ \text{S}_2\ \rrbracket\end{bmatrix}$
$\llbracket\ [_{\text{S}}\ \text{Neg S}]\ \rrbracket = \llbracket\ \text{Neg}\ \rrbracket(\llbracket\ \text{S}\ \rrbracket)$
$\llbracket\ [_{\text{S}}\ \text{NP VP}]\ \rrbracket = \llbracket\ \text{NP}\ \rrbracket(\llbracket\ \text{VP}\ \rrbracket)$
$\llbracket\ [_{\text{NP}}\ \text{Det N}]\ \rrbracket = \llbracket\ \text{Det}\ \rrbracket(\llbracket\ \text{N}\ \rrbracket)$
$\llbracket\ [_{\text{N}'}\ \text{Adj N}']\ \rrbracket = \llbracket\ \text{Adj}\ \rrbracket(\llbracket\ \text{N}'\ \rrbracket)$
$\llbracket\ [_{\text{NP}}\ \text{N}_{prop}]\ \rrbracket = \llbracket\ \text{N}_{prop}\ \rrbracket$
$\llbracket\ [_{\text{VP}}\ \text{V}_t\ \text{NP}]\ \rrbracket = \llbracket\ \text{V}_t\ \rrbracket(\llbracket\ \text{NP}\ \rrbracket)$
$\llbracket\ [_{\text{VP}}\ \text{V}_i]\ \rrbracket = \llbracket\ \text{V}_i\ \rrbracket$
$\llbracket\ [_{\text{Neg}}\ \text{“it's not true that”}]\ \rrbracket = \lambda T{:}\mathit{RecType}(\begin{bmatrix}\text{e}{:}\neg T\end{bmatrix})$
$\llbracket\ [_{\text{Det}}\ \text{“a”}]\ \rrbracket = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix}\text{x}{:}\mathit{Ind} \\ \text{restr}{:}Q(\begin{bmatrix}\text{x}{=}\Uparrow\text{x}\end{bmatrix}) \\ \text{e}{:}P(\begin{bmatrix}\text{x}{=}\Uparrow\text{x}\end{bmatrix})\end{bmatrix}))$
$\llbracket\ [_{\text{Det}}\ \text{“some”}]\ \rrbracket = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix}\text{e}{:}\text{some}(Q,\ P)\end{bmatrix}))$
$\llbracket\ [_{\text{Det}}\ \text{“every”}]\ \rrbracket = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix}\text{e}{:}\text{every}(Q,\ P)\end{bmatrix}))$
$\llbracket\ [_{\text{Det}}\ \text{“most”}]\ \rrbracket = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix}\text{e}{:}\text{most}(Q,\ P)\end{bmatrix}))$
$\llbracket\ [_{\text{N}}\ \text{“boy”}]\ \rrbracket = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e}{:}\text{boy}(r.\text{x})\end{bmatrix})$
$\llbracket\ [_{\text{N}}\ \text{“girl”}]\ \rrbracket = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e}{:}\text{girl}(r.\text{x})\end{bmatrix})$
$\llbracket\ [_{\text{Adj}}\ \text{“green”}]\ \rrbracket = \lambda P{:}\mathit{Ppty}(\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e}{:}\text{green}(r.\text{x},P)\end{bmatrix}))$
$\llbracket\ [_{\text{Adj}}\ \text{“imaginary”}]\ \rrbracket = \lambda P{:}\mathit{Ppty}(\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e}{:}\text{imaginary}(r.\text{x},P)\end{bmatrix}))$
$\llbracket\ [_{\text{N}_{prop}}\ \text{“Kim”}]\ \rrbracket = \lambda P{:}\mathit{Ppty}(P(\begin{bmatrix}\text{x}{=}\text{kim}\end{bmatrix}))$
$\llbracket\ [_{\text{N}_{prop}}\ \text{“Sandy”}]\ \rrbracket = \lambda P{:}\mathit{Ppty}(P(\begin{bmatrix}\text{x}{=}\text{sandy}\end{bmatrix}))$
$\llbracket\ [_{\text{V}_t}\ \text{“knows”}]\ \rrbracket = \lambda \mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\mathcal{P}(\lambda r_2{:}(\begin{bmatrix}\text{e}{:}\text{know}(r_1.\text{x},r_2.\text{x})\end{bmatrix}))))$
$\llbracket\ [_{\text{V}_t}\ \text{“sees”}]\ \rrbracket = \lambda \mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\mathcal{P}(\lambda r_2{:}(\begin{bmatrix}\text{e}{:}\text{see}(r_1.\text{x},r_2.\text{x})\end{bmatrix}))))$
$\llbracket\ [_{\text{V}_i}\ \text{“smiles”}]\ \rrbracket = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e}{:}\text{smile}(r.\text{x})\end{bmatrix})$
$\llbracket\ [_{\text{V}_i}\ \text{“laughs”}]\ \rrbracket = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e}{:}\text{laugh}(r.\text{x})\end{bmatrix})$

We have presented two alternative semantics for indefinites here: a

DRT style analysis (arbitrarily associated with the indefinite article "a") and a generalized quantifier analysis (arbitrarily associated with the determiner "some"). In the semantics for "a" we use a notation involving '$\Uparrow$'. Intuitively, the notation '$\Uparrow x$' refers to the 'x' in the next higher record. Recall that according to clause 3 of the definition of a dependent record type on page 17 a field which depends on another field for its value has in place of a type, an ordered pair consisting of a dependent type (a function from objects of some type to types) and a sequence of paths in the record type. The idea is that the dependent type should be applied to the objects found in the corresponding paths in the record which is being checked for membership of the record type. Thus a notation such as

$$Q(\left[\mathrm{x}{=}{\Uparrow}\mathrm{x}\right])$$

as in the semantics defined for "a" is an informal abbreviation for the pair

$$\langle \lambda v{:}\mathit{Ind}(Q(\left[\mathrm{x}{=}v\right])),\ \langle \mathrm{x}\rangle\rangle$$

If P is a property, that is $P{:}\mathit{Ppty}$, then let $[^{\tau}P]$ be the type of objects which have P, that is $a : [^{\tau}P]$ iff there is some b such that $b : P(\left[\mathrm{x}{=}a\right])$. If T is a type, let $[\check{}T]$ be $\{a \mid a : T\}$, the *extension* (or set of witnesses) of T. A model $\langle A, F\rangle$ for this language (that is an assignment A of objects to basic types, and an assignment F of sets of objects and situations to ptypes) is one according to which:

$\{\mathrm{kim, sandy}\} \subseteq A\ (\mathit{Ind})$
F is defined on the set of ptypes such that
$F(\mathrm{some}(Q{,}P)) \neq \emptyset$ iff $[\check{}[^{\tau}Q]] \cap [\check{}[^{\tau}P]] \neq \emptyset$
$F(\mathrm{every}(Q{,}P)) \neq \emptyset$ iff $[\check{}[^{\tau}Q]] \subseteq [\check{}[^{\tau}P]]$
$F(\mathrm{most}(Q{,}P)) \neq \emptyset$ iff $\frac{|[\check{}[^{\tau}Q]] \cap [\check{}[^{\tau}P]]|}{|[\check{}[^{\tau}Q]]|} > \theta_{most}$, where $[\check{}[^{\tau}Q]]$ is non-empty, and θ_{most} is a contextually determined parameter whose value is the threshold proportion of objects of type $[\check{}[^{\tau}Q]]$ that must be of type $[\check{}[^{\tau}P]]$ in order for it to be the case that most Q are P. If $[\check{}[^{\tau}Q]]$ is empty, then $F(\mathrm{most}(Q{,}P)) \neq \emptyset$ (assuming we want a "logical" interpretation where *most men run* is trivially true if there are no men).

Examples:

$[\![\ [_{\mathrm{S}}\ [_{\mathrm{NP}}\ [_{\mathrm{N}_{prop}}\ \mathrm{Kim}]]\ [_{\mathrm{VP}}\ [_{\mathrm{V}_i}\ \mathrm{smiles}]]]\ ]\!] =$
$\lambda P{:}\mathit{Ppty}(P(\left[\mathrm{x}{=}\mathrm{kim}\right]))(\lambda r{:}\left[\mathrm{x}{:}\mathit{Ind}\right](\left[\mathrm{e}{:}\mathrm{smile}(r.\mathrm{x})\right])) =$
$\lambda r{:}\left[\mathrm{x}{:}\mathit{Ind}\right](\left[\mathrm{e}{:}\mathrm{smile}(r.\mathrm{x})\right])(\left[\mathrm{x}{=}\mathrm{kim}\right]) =$
$\left[\mathrm{e}{:}\mathrm{smile}(\mathrm{kim})\right]$

$[\![\ [_{\mathrm{S}}\ [_{\mathrm{NP}}\ [_{\mathrm{Det}}\ \mathrm{a}]\ [_{\mathrm{N}}\ \mathrm{boy}]]\ [_{\mathrm{VP}}\ [_{\mathrm{V}_i}\ \mathrm{smiles}]]]\ ]\!] =$

$$\lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\ \begin{bmatrix}\text{x}:\mathit{Ind}\\ \text{restr}:Q(\bigl[\text{x}{=}{\Uparrow}\text{x}\bigr])\\ \text{e}:P(\bigl[\text{x}{=}{\Uparrow}\text{x}\bigr])\end{bmatrix}))$$
$$(\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:boy}(r.\text{x})\bigr]))$$
$$(\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:smile}(r.\text{x})\bigr])) =$$
$$\begin{bmatrix}\text{x}:\mathit{Ind}\\ \text{restr}:\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:boy}(r.\text{x})\bigr])(\bigl[\text{x}{=}{\Uparrow}\text{x}\bigr])\\ \text{e}:\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:smile}(r.\text{x})\bigr])(\bigl[\text{x}{=}{\Uparrow}\text{x}\bigr])\end{bmatrix} =$$
$$\begin{bmatrix}\text{x}:\mathit{Ind}\\ \text{restr}:\bigl[\text{e:boy}({\Uparrow}\text{x})\bigr]\\ \text{e}:\bigl[\text{e:smile}({\Uparrow}\text{x})\bigr]\end{bmatrix}$$

(In this example we have used the ⇑-notation introduced on p. 23.)

Note that any record of this last type will be (multiset) extensionally equivalent (in the sense defined in Cooper, 2012) to a record of the type

$$\begin{bmatrix}\text{x}:\mathit{Ind}\\ \text{restr:boy}(\text{x})\\ \text{e:smile}(\text{x})\end{bmatrix}$$

That is, the set (or multiset) of objects constituting the leaves of the records (i.e. the objects which are at the end of the paths in the records) will be identical. Although the records themselves are distinct because of the difference in the labels occurring on the paths, the commitment to constraints on the way that the world must be (represented by the objects which are the leaves) is the same. Thus the simplifications called *flattening* and *relabelling* in Cooper (2012) will preserve truth. That is, if T' is the result of flattening and/or relabelling T, then T' will be non-empty ("true") just in case T is non-empty. We will see below that these operations also preserve probability.

$[\![\ [_{\text{NP}}\ [_{\text{Det}}\ \text{every}]\ [_{\text{N}}\ \text{boy}]]\ ]\!] =$
$\lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\bigl[\text{e:every}(Q,\ P)\bigr]))(\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:boy}(r.\text{x})\bigr])) =$
$\lambda P{:}\mathit{Ppty}(\bigl[\text{e:every}(\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:boy}(r.\text{x})\bigr]),\ P)\bigr])$

$[\![\ [_{\text{VP}}\ [_{\text{V}_t}\ \text{knows}]\ [_{\text{NP}}\ [_{\text{N}_{prop}}\ \text{Sandy}]]]\ ]\!] =$
$\lambda\mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\mathcal{P}(\lambda r_2{:}(\bigl[\text{e:know}(r_1.\text{x},r_2.\text{x})\bigr]))))(\lambda P{:}\mathit{Ppty}(P(\bigl[\text{x=sandy}\bigr]))) =$
$\lambda r_1{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\lambda P{:}\mathit{Ppty}(P(\bigl[\text{x=sandy}\bigr]))(\lambda r_2{:}(\bigl[\text{e:know}(r_1.\text{x},r_2.\text{x})\bigr]))) =$
$\lambda r_1{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\lambda r_2{:}(\bigl[\text{e:know}(r_1.\text{x},r_2.\text{x})\bigr])(\bigl[\text{x=sandy}\bigr])) =$
$\lambda r_1{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:know}(r_1.\text{x,sandy})\bigr])$

$[\![\ [_{\text{S}}\ [_{\text{NP}}\ [_{\text{Det}}\ \text{every}]\ [_{\text{N}}\ \text{boy}]]\ [_{\text{VP}}\ [_{\text{V}_t}\ \text{knows}]\ [_{\text{NP}}\ [_{\text{N}_{prop}}\ \text{Sandy}]]]]\ ]\!] =$
$\lambda P{:}\mathit{Ppty}(\bigl[\text{e:every}(\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:boy}(r.\text{x})\bigr]),\ P)\bigr])(\lambda r_1{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:know}(r_1.\text{x,sandy})\bigr]))$
$=$
$\bigl[\text{e:every}(\lambda r{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:boy}(r.\text{x})\bigr]),\ \lambda r_1{:}\bigl[\text{x}{:}\mathit{Ind}\bigr](\bigl[\text{e:know}(r_1.\text{x,sandy})\bigr]))\bigr]$

$$[\![\ [_{\mathrm{S}}\ [_{\mathrm{S}}\ [_{\mathrm{NP}}\ [_{\mathrm{Det}}\ \text{every}]\ [_{\mathrm{N}}\ \text{boy}]]\ [_{\mathrm{VP}}\ [_{\mathrm{V}_t}\ \text{knows}]\ [_{\mathrm{NP}}\ [_{\mathrm{N}_{prop}}\ \text{Sandy}]]]]$$
$$\text{and}\ [_{\mathrm{S}}\ [_{\mathrm{NP}}\ [_{\mathrm{N}_{prop}}\ \text{Kim}]]\ [_{\mathrm{VP}}\ [_{\mathrm{V}_i}\ \text{smiles}]]]]\ ]\!] =$$
$$\left[\begin{array}{lll} \mathrm{e}_1 & : & \left[\text{e:every}(\lambda r{:}\left[\text{x:}\mathit{Ind}\right](\left[\text{e:boy}(r.\text{x})\right]),\ \lambda r_1{:}\left[\text{x:}\mathit{Ind}\right](\left[\text{e:know}(r_1.\text{x,sandy})\right]))\right] \\ \mathrm{e}_2 & : & \left[\text{e:smile(kim)}\right] \end{array}\right]$$

Note that any record of this last type will be (multiset) extensionally equivalent (in the sense defined in Cooper, 2012) to a record of the type

$$\left[\begin{array}{lll} \mathrm{e}_1 & : & \text{every}(\lambda r{:}\left[\text{x:}\mathit{Ind}\right](\left[\text{e:boy}(r.\text{x})\right]),\ \lambda r_1{:}\left[\text{x:}\mathit{Ind}\right](\left[\text{e:know}(r_1.\text{x,sandy})\right])) \\ \mathrm{e}_2 & : & \text{smile(kim)} \end{array}\right]$$

We have shown how a compositional semantics can associate a type with declarative sentences. According to this semantics the type will always be a record type representing a type of situations. Thus we can consider a type *Sit* ("situation") to be identical with *Rec* ("record"). An agent can then judge whether a situation, s, is the type, T, assigned to a sentence by our semantics. Such a judgement is normally not considered as a type theoretic object. However, Ginzburg (2012) suggests that such judgements can correspond to type theoretic objects called Austinian propositions (which have their origins in the work of Barwise and Perry (1983) on situation semantics, where Austinian propositions are introduced on the basis of Austin (1961)). Austin's idea was that propositions are not objects that are true or false *simpliciter* but are true or false with respect to the part of the world that is being described. Barwise and Perry took their situations to be part of the world and what they called Austinian propositions as pairings of a situation and a situation type.

Ginzburg's proposal is to use Austinian propositions as records of the type

$$\left[\begin{array}{lll} \text{sit} & : & \mathit{Sit} \\ \text{sit-type} & : & \mathit{Type} \end{array}\right]$$

A record of this type, say,

$$\left[\begin{array}{lll} \text{sit} & = & s \\ \text{sit-type} & = & T \end{array}\right]$$

is an Austinian proposition which associates the situation s with the type T. It uses the labels 'sit' and 'sit-type' to allow us to access the components of the Austinian proposition. It is a strategy which uses records in the same way as we use ordered pairs like $\langle s, T\rangle$ in standard set theory, except that we have explicit labelling, rather than an ordering, to recover the components.

We take the type *Sit* ("situation") to be identical with the type *Rec* ("record") where $r : \mathit{Rec}$ iff r is a record as defined in the section on record types above. The leading idea is that such a proposition ϕ is true just in case ϕ.sit : ϕ.sit-type.

It is then trivial to recast our grammar so that it maps expressions of the language and a situation, s, to an Austinian proposition which may be regarded as the content of the expression for a given described situation:[14]

$$\llbracket\ [_{\text{S}}\ \text{S}_1 \text{ and } \text{S}_2]\ \rrbracket^s = \begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e}_1{:}\llbracket\ \text{S}_1\ \rrbracket \\ \text{e}_2{:}\llbracket\ \text{S}_2\ \rrbracket\end{bmatrix}\end{bmatrix}$$

$$\llbracket\ [_{\text{S}}\ \text{S}_1 \text{ or } \text{S}_2]\ \rrbracket^s = \begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e}{:}\llbracket\ \text{S}_1\ \rrbracket \vee \llbracket\ \text{S}_2\ \rrbracket\end{bmatrix}\end{bmatrix}$$

$$\llbracket\ [_{\text{S}}\ \text{Neg S}]\ \rrbracket^s = \llbracket\ \text{Neg}\ \rrbracket^s(\llbracket\ \text{S}\ \rrbracket)$$

$$\llbracket\ [_{\text{S}}\ \text{NP VP}]\ \rrbracket^s = \llbracket\ \text{NP}\ \rrbracket^s(\llbracket\ \text{VP}\ \rrbracket)$$

$$\llbracket\ [_{\text{NP}}\ \text{Det N}]\ \rrbracket^s = \llbracket\ \text{Det}\ \rrbracket^s(\llbracket\ \text{N}\ \rrbracket)$$

$$\llbracket\ [_{\text{N}'}\ \text{Adj N}']\ \rrbracket^s = \llbracket\ \text{Adj}\ \rrbracket^s(\llbracket\ \text{N}'\ \rrbracket)$$

$$\llbracket\ [_{\text{NP}}\ \text{N}_{prop}]\ \rrbracket^s = \llbracket\ \text{N}_{prop}\ \rrbracket^s$$

$$\llbracket\ [_{\text{VP}}\ \text{V}_t\ \text{NP}]\ \rrbracket^s = \llbracket\ \text{V}_t\ \rrbracket^s(\llbracket\ \text{NP}\ \rrbracket)$$

$$\llbracket\ [_{\text{VP}}\ \text{V}_i]\ \rrbracket^s = \llbracket\ \text{V}_i\ \rrbracket^s$$

$$\llbracket\ [_{\text{Neg}}\ \text{“it's not true that”}]\ \rrbracket^s = \lambda T{:}\mathit{RecType}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e}{:}\neg T\end{bmatrix}\end{bmatrix})$$

$$\llbracket\ [_{\text{Det}}\ \text{“a”}]\ \rrbracket^s = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{x}{:}\mathit{Ind} \\ \text{restr}{:}Q(\begin{bmatrix}\text{x}{=}\Uparrow\text{x}\end{bmatrix}) \\ \text{e}{:}P(\begin{bmatrix}\text{x}{=}\Uparrow\text{x}\end{bmatrix})\end{bmatrix}\end{bmatrix})$$

$$\llbracket\ [_{\text{Det}}\ \text{“some”}]\ \rrbracket^s = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:some}(Q,\ P)\end{bmatrix}\end{bmatrix}))$$

$$\llbracket\ [_{\text{Det}}\ \text{“every”}]\ \rrbracket^s = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:every}(Q,\ P)\end{bmatrix}\end{bmatrix}))$$

$$\llbracket\ [_{\text{Det}}\ \text{“most”}]\ \rrbracket^s = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:most}(Q,\ P)\end{bmatrix}\end{bmatrix}))$$

$$\llbracket\ [_{\text{N}}\ \text{“boy”}]\ \rrbracket^s = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:boy}(r.\text{x})\end{bmatrix}\end{bmatrix})$$

$$\llbracket\ [_{\text{N}}\ \text{“girl”}]\ \rrbracket^s = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:girl}(r.\text{x})\end{bmatrix}\end{bmatrix})$$

$$\llbracket\ [_{\text{Adj}}\ \text{“green”}]\ \rrbracket^s = \lambda P{:}\mathit{Ppty}(\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:green}(r.\text{x},P)\end{bmatrix}\end{bmatrix}))$$

$$\llbracket\ [_{\text{Adj}}\ \text{“imaginary”}]\ \rrbracket^s = \lambda P{:}\mathit{Ppty}(\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:imaginary}(r.\text{x},P)\end{bmatrix}\end{bmatrix}))$$

$$\llbracket\ [_{\text{N}_{prop}}\ \text{“Kim”}]\ \rrbracket^s = \lambda P{:}\mathit{Ppty}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & P(\begin{bmatrix}\text{x}{=}\text{kim}\end{bmatrix})\end{bmatrix})$$

$$\llbracket\ [_{\text{N}_{prop}}\ \text{“Sandy”}]\ \rrbracket^s = \lambda P{:}\mathit{Ppty}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & P(\begin{bmatrix}\text{x}{=}\text{sandy}\end{bmatrix})\end{bmatrix})$$

$$\llbracket\ [_{\text{V}_t}\ \text{“knows”}]\ \rrbracket^s = \lambda\mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\mathcal{P}(\lambda r_2{:}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:know}(r_1.\text{x},r_2.\text{x})\end{bmatrix}\end{bmatrix}))))$$

$$\llbracket\ [_{\text{V}_t}\ \text{“sees”}]\ \rrbracket^s = \lambda\mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\mathcal{P}(\lambda r_2{:}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:see}(r_1.\text{x},r_2.\text{x})\end{bmatrix}\end{bmatrix}))))$$

$$\llbracket\ [_{\text{V}_i}\ \text{“smiles”}]\ \rrbracket^s = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix}\text{e:smile}(r.\text{x})\end{bmatrix}\end{bmatrix})$$

[14] Here we use $\llbracket\ \alpha\ \rrbracket$ to represent the content generated by the previous grammar.

$$[\![\; [_{\mathrm{V}_i} \text{“laughs”}] \;]\!]^s = \lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big]\left(\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \big[\text{e:laugh}(r.\mathrm{x})\big] \end{bmatrix}\right)$$

3.2 Interpretation with Probabilistic Types

Montague (1974) determines the denotation of a complex expression by applying a function to an intensional argument (as in $[\![\text{ NP }]\!](^\wedge[\![\text{ VP }]\!])$). We employ a variant of this general strategy by applying a probabilistic evaluation function $[\![\cdot]\!]_p$ to a categorical (non-probabilistic) semantic value. For semantic categories that are interpreted as functions, $[\![\cdot]\!]_p$ returns functions from categorical values to probabilistic functions, or to probabilities. For sentences it produces probability values.

The probabilistic evaluation function $[\![\cdot]\!]_p$ produces a probabilistic interpretation based on the compositional semantics given in Section 3.1. For sentences it will return the probability that the sentence is true. We are not proposing strict compositionality in terms of probabilities. Probabilities are like truth-values (or rather, truth-values are the limit cases of probabilities). We would not expect to be able to compute the probability associated with a complex constituent on the basis of the probabilities associated with its immediate constituents any more than we would expect to be able to compute a categorical interpretation entirely in terms of truth-functions and extensions. However, the simultaneous computation of categorical and probabilistic interpretations provides us with a compositional semantic system that is closely related to the simultaneous computation of intensions and extensions in classical Montague semantics.

$$[\![\; [_{\mathrm{S}} \; \mathrm{S}_1 \text{ and } \mathrm{S}_2] \;]\!]_p = \mathrm{p}\left(\begin{bmatrix} \mathrm{e}_1{:}[\![\text{ S}_1 \text{ }]\!] \\ \mathrm{e}_2{:}[\![\text{ S}_2 \text{ }]\!] \end{bmatrix}\right),$$

that is, the probability that there is something of type $\begin{bmatrix} \mathrm{e}_1{:}[\![\text{ S}_1 \text{ }]\!] \\ \mathrm{e}_2{:}[\![\text{ S}_2 \text{ }]\!] \end{bmatrix}$.

$[\![\; [_{\mathrm{S}} \; \mathrm{S}_1 \text{ or } \mathrm{S}_2] \;]\!]_p = \mathrm{p}(\big[\mathrm{e}{:}[\![\text{ S}_1 \text{ }]\!] \vee [\![\text{ S}_2 \text{ }]\!]\big])$
$[\![\; [_{\mathrm{S}} \text{ Neg S}] \;]\!]_p = [\![\text{ Neg }]\!]_p([\![\text{ S }]\!])$
$[\![\; [_{\mathrm{S}} \text{ NP VP}] \;]\!]_p = [\![\text{ NP }]\!]_p([\![\text{ VP }]\!])$
$[\![\; [_{\mathrm{NP}} \text{ Det N}] \;]\!]_p = [\![\text{ Det }]\!]_p([\![\text{ N }]\!])$
$[\![\; [_{\mathrm{N}'} \text{ Adj N}'] \;]\!]_p = [\![\text{ Adj }]\!]_p([\![\text{ N}' \text{ }]\!])$
$[\![\; [_{\mathrm{NP}} \; \mathrm{N}_{\mathit{prop}}] \;]\!]_p = [\![\; \mathrm{N}_{\mathit{prop}} \;]\!]_p$
$[\![\; [_{\mathrm{VP}} \; \mathrm{V}_t \text{ NP}] \;]\!]_p = [\![\; \mathrm{V}_t \;]\!]_p([\![\text{ NP }]\!])$
$[\![\; [_{\mathrm{VP}} \; \mathrm{V}_i] \;]\!]_p = [\![\; \mathrm{V}_i \;]\!]_p$

$[\![\; [_{\mathrm{Neg}} \text{ “it's not true that”}] \;]\!]_p = \lambda T{:}\mathit{RecType}(\mathrm{p}(\big[\mathrm{e}{:}\neg T\big]))$
$[\![\; [_{\mathrm{Det}} \text{ “some”}] \;]\!]_p = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\mathrm{p}(\big[\text{e:some}(Q,\, P)\big])))$
$[\![\; [_{\mathrm{Det}} \text{ “every”}] \;]\!]_p = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\mathrm{p}(\big[\text{e:every}(Q,\, P)\big])))$
$[\![\; [_{\mathrm{Det}} \text{ “most”}] \;]\!]_p = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\mathrm{p}(\big[\text{e:most}(Q,\, P)\big])))$
$[\![\; [_{\mathrm{N}} \text{ “boy”}] \;]\!]_p = \lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}(\big[\text{e:boy}(r.\mathrm{x})\big]))$

$\llbracket\ [_{\mathrm{N}}\ \text{“girl”}]\ \rrbracket_p = \lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}(\big[\mathrm{e}{:}\mathrm{girl}(r.\mathrm{x})\big]))$

$\llbracket\ [_{\mathrm{Adj}}\ \text{“green”}]\ \rrbracket_p = \lambda P{:}\mathit{Ppty}(\lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}((\big[\mathrm{e}{:}\mathrm{green}(r.\mathrm{x},P)\big])))))$

$\llbracket\ [_{\mathrm{Adj}}\ \text{“imaginary”}]\ \rrbracket_p = \lambda P{:}\mathit{Ppty}(\lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}((\big[\mathrm{e}{:}\mathrm{imaginary}(r.\mathrm{x},P)\big])))))$

$\llbracket\ [_{\mathrm{N}_{prop}}\ \text{“Kim”}]\ \rrbracket_p = \lambda P{:}\mathit{Ppty}(\mathrm{p}(P(\big[\mathrm{x}{=}\mathrm{kim}\big])))$

$\llbracket\ [_{\mathrm{N}_{prop}}\ \text{“Sandy”}]\ \rrbracket_p = \lambda P{:}\mathit{Ppty}(\mathrm{p}(P(\big[\mathrm{x}{=}\mathrm{sandy}\big])))$

$\llbracket\ [_{\mathrm{V}_t}\ \text{“knows”}]\ \rrbracket_p = \lambda \mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}(\mathcal{P}(\lambda r_2{:}(\big[\mathrm{e}{:}\mathrm{know}(r_1.\mathrm{x},r_2.\mathrm{x})\big])))))$

$\llbracket\ [_{\mathrm{V}_t}\ \text{“sees”}]\ \rrbracket_p = \lambda \mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}(\mathcal{P}(\lambda r_2{:}(\big[\mathrm{e}{:}\mathrm{see}(r_1.\mathrm{x},r_2.\mathrm{x})\big])))))$

$\llbracket\ [_{\mathrm{V}_i}\ \text{“smiles”}]\ \rrbracket_p = \lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}(\big[\mathrm{e}{:}\mathrm{smile}(r.\mathrm{x})\big]))$

$\llbracket\ [_{\mathrm{V}_i}\ \text{“laughs”}]\ \rrbracket_p = \lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\mathrm{p}(\big[\mathrm{e}{:}\mathrm{laugh}(r.\mathrm{x})\big]))$

A probability distribution d for this language, based on a set of situations $\mathcal{S}$, is such that:

$p_d(a : \mathit{Ind}) = 1$ if a is kim or sandy[15]

$p_d(s : T) \in [0,1]$ if $s \in \mathcal{S}$ and T is a ptype

$p_d(s : T) = 0$ if $s \notin \mathcal{S}$ and T is a ptype[16]

$p_d(a : [^\tau P]) = p_d(P(\big[\mathrm{x}{=}a\big]))$

$p_d(\mathrm{some}(P,Q)) = p_d([^\tau P] \wedge [^\tau Q])$

$p_d(\mathrm{every}(P,Q)) = p_d([^\tau P] \rightarrow [^\tau Q])$

$p_d(\mathrm{most}(P,Q)) = \min(1, \frac{p_d([^\tau P] \wedge [^\tau Q])}{\theta_{most}\ p_d([^\tau P])})$

The probability that an event e is of the type in which the relation *some* holds of the properties P and Q is the probability that e is of the conjunctive type $P \wedge Q$. The probability that e is of the *every* type for P and Q is the likelihood that it instantiates the functional type $P \rightarrow Q$. As we have defined the probabilities associated with functional types in terms of universal quantification (an unbounded conjunction of the pairings between the elements of the domain P of the function and its range Q), this definition sustains the desired reading of *every.* The likelihood that e is of the type *most* for P and Q is the likelihood that e is of type $P \wedge Q$, factored by the product of the contextually determined parameter θ_{most} and the likelihood that e is of type P, where this fraction is less than 1, and 1 otherwise. This account of *most* approximates the categorical interpretation of *most* given in Section 3.1, with probabilities replacing cardinality values.

Let's consider a simple example.

Example:

$\llbracket\ [_{\mathrm{S}}\ [_{\mathrm{NP}}\ [_{\mathrm{N}_{prop}}\ \mathrm{Kim}]]\ [_{\mathrm{VP}}\ [_{\mathrm{V}_i}\ \mathrm{smiles}]]]\ \rrbracket_p =$

$\lambda P{:}\mathit{Ppty}(\mathrm{p}(P(\big[\mathrm{x}{=}\mathrm{kim}\big])))(\lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\big[\mathrm{e}{:}\mathrm{smile}(r.\mathrm{x})\big])) =$

$\mathrm{p}(\lambda r{:}\big[\mathrm{x}{:}\mathit{Ind}\big](\big[\mathrm{e}{:}\mathrm{smile}(r.\mathrm{x})\big])(\big[\mathrm{x}{=}\mathrm{kim}\big])) =$

[15] This seems an intuitive assumption, though not a necessary one.

[16] Again this seems an intuitive, though not a necessary assumption.

$\mathrm{p}(\left[\text{e:smile(kim)}\right])$

Suppose that

$$p_d(s_1\text{:smile(kim)})\text{=}.7$$
$$p_d(s_2\text{:smile(kim)})\text{=}.3$$
$$p_d(s_3\text{:smile(kim)})\text{=}.4$$

and there are no other situations s_i such that $p_d(s_i\text{:smile(kim)})>0$. Furthermore, let us assume that these probabilities are independent of each other, that is, $p_d(s_3\text{:smile(kim)}) = p_d(s_3\text{:smile(kim)} \mid s_1\text{:smile(kim)}, s_2\text{:smile(kim)})$ and so on. Then

$$p_d(\text{smile(kim)})\text{=}$$
$$\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}, s_3 : \text{smile(kim)})\text{=}$$
$$\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}) + .4 - .4\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)})\text{=}$$
$$(.7 + .3 - .7 \times .3) + .4 - .4(.7 + .3 - .7 \times .3)\text{=}$$
$$.874$$

This means that $p_d(\left[\text{e:smile(kim)}\right]) = .874$.

Hence $\llbracket\ [_{\text{S}}\ [_{\text{NP}}\ [_{\text{N}_{prop}}\ \text{Kim}]]\ [_{\text{VP}}\ [_{\text{V}_i}\ \text{smiles}]]]\ \rrbracket_{p_d} = .874$ (where $\llbracket\ \alpha\ \rrbracket_{p_d}$ is the result of computing $\llbracket\ \alpha\ \rrbracket_p$ with respect to the probability distribution d).

The probabilistic semantics we have presented above will compute for a sentence α the probability that there is something of the type (of situation) which constitutes the content of α in terms of the probabilistic and the categorical interpretations associated with α's constituents. If the interpretation of α, $\llbracket\ \alpha\ \rrbracket$, is a type T, then $\llbracket\ \alpha\ \rrbracket_p$ is $p_d(T)$, where d is a given probability function which assigns probabilities to basic types and ptypes. A more explicit notation would be $\llbracket\ \alpha\ \rrbracket_p^d$, which makes it clear that we are computing a probability with respect to a probability distribution that represents a view of (part of) the world.

This makes our semantics behave like a truth-conditional semantics, but instead of using truth values in the set $\{0, 1\}$, we situate the value of a sentence in the interval $[0, 1]$. Saying that $\llbracket\ \alpha\ \rrbracket_p^d = p$ means that the probability that there is a situation that makes α true is p, ie. α is true to degree p. This, then, is a probabilistic variant of a classical semantics based on truth-conditions. For a sentence α it answers the question: "what is the likelihood that α is true?".

However, with the tools that we have developed, we could also answer another question: given a particular situation e and a sentence α, "what is the probability that e is of the type $\llbracket\ \alpha\ \rrbracket$ with respect to a given distribution d?". This is a question about $p_d(e : T)$, where T is $\llbracket\ \alpha\ \rrbracket$. It is straightforward to define this in a way similar to our previously specified probabilistic semantics. As before, we suppress the distribution d.

$$\llbracket\ [_{\text{S}}\ \text{S}_1 \text{ and } \text{S}_2]\ \rrbracket^e_p = \text{p}(e : \begin{bmatrix}\text{e}_1{:}\llbracket\ \text{S}_1\ \rrbracket \\ \text{e}_2{:}\llbracket\ \text{S}_2\ \rrbracket\end{bmatrix})$$
$$\llbracket\ [_{\text{S}}\ \text{S}_1 \text{ or } \text{S}_2]\ \rrbracket^e_p = \text{p}(e : \begin{bmatrix}\text{e}{:}\llbracket\ \text{S}_1\ \rrbracket \vee \llbracket\ \text{S}_2\ \rrbracket\end{bmatrix})$$
$$\llbracket\ [_{\text{S}}\ \text{Neg S}]\ \rrbracket^e_p = \llbracket\ \text{Neg}\ \rrbracket^e_p(\llbracket\ \text{S}\ \rrbracket)$$
$$\llbracket\ [_{\text{S}}\ \text{NP VP}]\ \rrbracket^e_p = \llbracket\ \text{NP}\ \rrbracket^e_p(\llbracket\ \text{VP}\ \rrbracket)$$
$$\llbracket\ [_{\text{NP}}\ \text{Det N}]\ \rrbracket^e_p = \llbracket\ \text{Det}\ \rrbracket^e_p(\llbracket\ \text{N}\ \rrbracket)$$
$$\llbracket\ [_{\text{N}'}\ \text{Adj N}']\ \rrbracket^e_p = \llbracket\ \text{Adj}\ \rrbracket^e_p(\llbracket\ \text{N}'\ \rrbracket)$$
$$\llbracket\ [_{\text{NP}}\ \text{N}_{prop}]\ \rrbracket^e_p = \llbracket\ \text{N}_{prop}\ \rrbracket^e_p$$
$$\llbracket\ [_{\text{VP}}\ \text{V}_t \text{ NP}]\ \rrbracket^e_p = \llbracket\ \text{V}_t\ \rrbracket^e_p(\llbracket\ \text{NP}\ \rrbracket)$$
$$\llbracket\ [_{\text{VP}}\ \text{V}_i]\ \rrbracket^e_p = \llbracket\ \text{V}_i\ \rrbracket^e_p$$

$$\llbracket\ [_{\text{Neg}}\ \text{“it's not true that”}]\ \rrbracket^e_p = \lambda T{:}\mathit{RecType}(\text{p}(e : \begin{bmatrix}\text{e}{:}\neg T\end{bmatrix}))$$
$$\llbracket\ [_{\text{Det}}\ \text{“some”}]\ \rrbracket^e_p = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{some}(Q,\ P)\end{bmatrix})))$$
$$\llbracket\ [_{\text{Det}}\ \text{“every”}]\ \rrbracket^e_p = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{every}(Q,\ P)\end{bmatrix})))$$
$$\llbracket\ [_{\text{Det}}\ \text{“most”}]\ \rrbracket^e_p = \lambda Q{:}\mathit{Ppty}(\lambda P{:}\mathit{Ppty}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{most}(Q,\ P)\end{bmatrix})))$$
$$\llbracket\ [_{\text{N}}\ \text{“boy”}]\ \rrbracket^e_p = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{boy}(r.\text{x})\end{bmatrix}))$$
$$\llbracket\ [_{\text{N}}\ \text{“girl”}]\ \rrbracket^e_p = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{girl}(r.\text{x})\end{bmatrix}))$$
$$\llbracket\ [_{\text{Adj}}\ \text{“green”}]\ \rrbracket^e_p = \lambda P{:}\mathit{Ppty}(\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}((e : \begin{bmatrix}\text{e}{:}\text{green}(r.\text{x},P)\end{bmatrix})))))$$
$$\llbracket\ [_{\text{Adj}}\ \text{“imaginary”}]\ \rrbracket^e_p = \lambda P{:}\mathit{Ppty}(\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}((e : \begin{bmatrix}\text{e}{:}\text{imaginary}(r.\text{x},P)\end{bmatrix})))))$$
$$\llbracket\ [_{\text{N}_{prop}}\ \text{“Kim”}]\ \rrbracket^e_p = \lambda P{:}\mathit{Ppty}(\text{p}(e : P(\begin{bmatrix}\text{x}{=}\text{kim}\end{bmatrix})))$$
$$\llbracket\ [_{\text{N}_{prop}}\ \text{“Sandy”}]\ \rrbracket^e_p = \lambda P{:}\mathit{Ppty}(\text{p}(e : P(\begin{bmatrix}\text{x}{=}\text{sandy}\end{bmatrix})))$$
$$\llbracket\ [_{\text{V}_t}\ \text{“knows”}]\ \rrbracket^e_p = \lambda \mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}(e : \mathcal{P}(\lambda r_2{:}(\begin{bmatrix}\text{e}{:}\text{know}(r_1.\text{x},r_2.\text{x})\end{bmatrix})))))$$
$$\llbracket\ [_{\text{V}_t}\ \text{“sees”}]\ \rrbracket^e_p = \lambda \mathcal{P}{:}\mathit{Quant}(\lambda r_1{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}(e : \mathcal{P}(\lambda r_2{:}(\begin{bmatrix}\text{e}{:}\text{see}(r_1.\text{x},r_2.\text{x})\end{bmatrix})))))$$
$$\llbracket\ [_{\text{V}_i}\ \text{“smiles”}]\ \rrbracket^e_p = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{smile}(r.\text{x})\end{bmatrix}))$$
$$\llbracket\ [_{\text{V}_i}\ \text{“laughs”}]\ \rrbracket^e_p = \lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}(\text{p}(e : \begin{bmatrix}\text{e}{:}\text{laugh}(r.\text{x})\end{bmatrix}))$$

This version of the semantics will be important for the learning theory that we discuss in section 4. We will be considering an agent who is confronted with a particular situation and must estimate the probability that it belongs to the type represented by a given natural language expression.

Just as for categorical semantics, we can construct type theoretic objects corresponding to probabilistic judgements. We will call these probabilistic Austinian propositions. These are records of type:

$$\begin{bmatrix}\text{sit} & : & \mathit{Sit} \\ \text{sit-type} & : & \mathit{Type} \\ \text{prob} & : & [0,1]\end{bmatrix}$$

where [0,1] is used to represent the type of real numbers between 0 and 1. In our learning theory probabilistic Austinian propositions give our agents a way of keeping a record of the past judgements which are important for estimating the probability of a new judgement. The grammar can be recast so that it returns probabilistic Austinian propositions in a similar fashion to the way in which we introduced Austinian propositions before. We give the first rule of the grammar as an example.

$$\llbracket\ [_{\mathrm{S}}\ \mathrm{S}_1 \text{ and } \mathrm{S}_2]\ \rrbracket^{s}_{p} = \begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \begin{bmatrix} \mathrm{e}_1{:}\llbracket\ \mathrm{S}_1\ \rrbracket \\ \mathrm{e}_2{:}\llbracket\ \mathrm{S}_2\ \rrbracket \end{bmatrix} \\ \text{prob} & = & \mathrm{p}(s{:}\begin{bmatrix} \mathrm{e}_1{:}\llbracket\ \mathrm{S}_1\ \rrbracket \\ \mathrm{e}_2{:}\llbracket\ \mathrm{S}_2\ \rrbracket \end{bmatrix}) \end{bmatrix}$$

4 Learning through observation

In Section 2 we introduced the basic machinery of probabilistic TTR, and in Section 3 we showed how a compositional semantics could yield probabilistic judgements. In this account we relied on the availability of various conditional probabilities computed from a model consisting of probabilistic judgements corresponding to probabilistic Austinian propositions. The task of the learning component of the theory is threefold. First, it explains how to compute conditional probabilities based on a model. Second, it shows how conditional probabilities derived from a model can be used to make new probabilistic judgements. Third, it specifies how a model can be learned from observations.

In this section, we will outline a learning theory for the probabilistic type judgements discussed in the earlier sections. The basic idea is that an agent makes probabilistic observations in the world, based on perceiving situations, and interacting with the world and other agents. These judgements can then be used for classifying new situations, thus yielding new probabilistic observations from which probabilistic judgements can be derived.

The learning theory presented here is limited to learning (an extension of) the standard Naive Bayes classifier, which of course makes it quite limited in scope. This is a simplification adopted here for ease of exposition, and should be regarded as an initial step towards a more complete learning theory. It is an illustrative fragment of larger project. This project involves specifying a connection between our probabilistic type theory and a Bayesian network of beliefs concerning semantically related classifier applications.

4.1 Learning and classification

We will assume that agents have a way of perceiving situations in the world (such situations include individuals and states of affairs). Apart from perceiving situations (in the sense of obtaining low-level sensory data from sense organs), agents are also able to detect individuals presented, and to classify properties of, and relations among individuals. Larsson (2013) shows how a perceptron classifier of real-valued perceptual input can be integrated into TTR. Fernández and Larsson (2014) gives an account of vagueness using a Bayesian classifier taking real-valued perceptual information as input.

Observing a situation yields an object of type *Sit*, the type of situations. *Sit* is thus the supertype for more specific types of situations. We assume that agents keep records of observed situations and their types, modelled as probabilistic Austinian propositions. For example, an observation of a situation s_1 involving a boy smiling might yield the following Austinian proposition:

$$\begin{bmatrix} \text{sit} & = & s_1 \\ \text{sit-type} & = & \begin{bmatrix} \text{x} & : & \mathit{Ind} \\ \text{c}_{\text{boy}} & : & \text{boy(x)} \\ \text{c}_{\text{smile}} & : & \text{smile(x)} \end{bmatrix} \\ \text{prob} & = & 0.7 \end{bmatrix}$$

An agent, A, will make judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$, corresponding to prior judgements held in memory. For a type, T, we will use $\mathfrak{J}_T$ to represent that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$:

$$\mathfrak{J}_T = \{j \mid j \in \mathfrak{J}, j.\text{sit-type} = T\}$$

If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions we use $\| T \|_{\mathfrak{J}}$ to represent the sum of all probabilities associated with T in $\mathfrak{J}$:

$$\| T \|_{\mathfrak{J}} = \sum_{j \in \mathfrak{J}_T} j.\text{prob}$$

We use $\Sigma(\mathfrak{J})$ to represent the total number of situations classified[17], that is,

$$\Sigma(\mathfrak{J}) = \mid \{s \mid j \in \mathfrak{J}, j.\mathit{sit} = s\} \mid$$

[17] This is one of several possible definitions of $\Sigma(\mathfrak{J})$, and the one that seems to be closest to the standard Bayesian formulation of priors. Other definitions include the sum of all probabilities in $\mathfrak{J}$, that is,

$$\Sigma(\mathfrak{J}) = \sum_{j \in \mathfrak{J}} j.\text{prob}$$

We leave the investigation and discussion of these alternatives to future work.

We will use $\text{prior}_{\mathfrak{J}}(T)$ to represent the prior probability that anything is of type T given $\mathfrak{J}$, that is,

$$\text{prior}_{\mathfrak{J}}(T) = \frac{|| \, T \, ||_{\mathfrak{J}}}{\Sigma(\mathfrak{J})}$$

if $\Sigma(\mathfrak{J}) > 0$, and 0 otherwise.

We will use the notation $\text{p}_{A,\mathfrak{J}}(s : T)$ (and similarly, $\text{p}_{A,\mathfrak{J}}(s : T_1 \mid s : T_2)$) to denote the probability that agent A assigns, with respect to prior judgements $\mathfrak{J}$, to s being of type T (and similarly, the probability that agent A assigns, with respect to prior judgements $\mathfrak{J}$, to s being of type T_1, given that A judges s to be of type T_2).[18]

Suppose agent A is confronted with a new situation s and considers whether it is of type T. Observing the situation amounts to using probabilistic reasoning to figure out the probability $\text{p}_{A,\mathfrak{J}}(s : T)$. Probabilistic reasoning crucially depends on conditional probabilities computed by inference from previous judgements.

Conditional probabilities are computed as follows:

$$\text{p}_{A,\mathfrak{J}}(s : T_1 \mid s : T_2) = \frac{|| \, T_1 \wedge T_2 \, ||_{\mathfrak{J}}}{|| \, T_2 \, ||_{\mathfrak{J}}}, \text{ if } || \, T_2 \, ||_{\mathfrak{J}} \neq 0$$

Otherwise,

$$\text{p}_{A,\mathfrak{J}}(s : T_1 \mid s : T_2) = 0$$

This is a TTR variant of the standard Bayesian formula for computing conditional probabilities:

$$\text{p}(A \mid B) = \frac{| \, A \& B \, |}{| \, B \, |}$$

In Section 1.2 we rendered the TTR variant thus:

$$\frac{|\{s \mid s : T_1 \wedge T_2\}|}{|\{s \mid s : T_2\}|}.$$

Here, however, instead of counting categorical judgements, we are summing probabilities of judgements. The reason for this is that our "training data" is not limited to categorical observations. Instead we assume that it consists of probabilistic observations of situations being of certain types, with these observations having specified probabilities. By using an observer's previous judgements on the probability of an event being of a particular type, as the prior for the rule that computes the

[18] We introduce the agent A for intuitive clarity, although here the probabilities associated with judgements depend only on the string of previous judgements $\mathfrak{J}$. Insofar as such strings of judgements are always specific to an agent, indexing probabilities with both agent and string is in principle superfluous.

probability of a new event being of that type, we have, in effect, compressed information that properly belongs in a Bayesian network into our characterisation of a Bayesian classifier. We will here refer to this "hybrid" as a *TTR Bayes classifier.* This is a simplification that we adopt here for clarity of explanation. As we have indicated above, in future work we will characterise classifier learning through full Bayesian networks.

To illustrate our approach, assume that we have the following types:

$$\mathrm{T}_{\mathrm{boy}} = \left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{boy}} & : & \mathrm{boy(x)} \end{array}\right] \text{ and}$$

$$\mathrm{T}_{\mathrm{smile}} = \left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{smile}} & : & \mathrm{smile(x)} \end{array}\right]$$

Assume also that $\mathfrak{J}_{T_{\mathrm{boy}} \wedge T_{\mathrm{smile}}}$ has three members (corresponding to judgements by A that a boy was smiling in three observed situations s_1, s_3 and s_4), and that these Austinian propositions have the probabilities 0.6, 0.6 and 0.5 respectively.

We take $\mathfrak{J}_{T_{\mathrm{boy}}}$ to have five members corresponding to judgements by A that there was a boy in $s_1, \ldots, s_5$, and that the Austinian propositions assigning $\mathrm{T}_{\mathrm{boy}}$ to $s_1, \ldots, s_5$ all have probability 0.7. With these assumptions, the conditional probability that A will assign on the basis of $\mathfrak{J}$ to someone smiles, given that he is a boy is

$$\mathrm{p}_{A,\mathfrak{J}}(s : \mathrm{T}_{\mathrm{smile}} \mid s : \mathrm{T}_{\mathrm{boy}}) = \frac{\| \, T_{\mathrm{boy}} \wedge T_{\mathrm{smile}} \, \|_{\mathfrak{J}}}{\| \, T_{\mathrm{boy}} \, \|_{\mathfrak{J}}} =$$

$$\frac{0.6 + 0.6 + 0.5}{0.7 + 0.7 + 0.7 + 0.7 + 0.7} = .486$$

Conditional probabilities can be used in a TTR Bayes classifier. Here, the idea is that A classifies a new situation s based on the prior judgements $\mathfrak{J}$, and whatever information A can acquire about s – the *evidence.* The evidence has the form $\mathrm{p}_{A,\mathfrak{J}}(s : T_{e_1})$, …, $\mathrm{p}_{A,\mathfrak{J}}(s : T_{e_n})$, where $T_{e_1}, \ldots, T_{e_n}$ are the *evidence types.* Correspondingly, associated with a classifier κ is a collection of evidence types $T^{\kappa}_{e_1}, T^{\kappa}_{e_2}, \ldots, T^{\kappa}_{e_n}$ and a collection of possible conclusion types $T^{\kappa}_{c_1}, T^{\kappa}_{c_2}, \ldots, T^{\kappa}_{c_m}$. The evidence is acquired by observing and classifying the situation s with respect to the evidence types. This can be done using another layer of probabilistic classification based on yet another layer of evidence types, or by probabilistic or non-probabilistic classification of low-level sensory readings resulting directly from observations (Larsson, 2013, Fernández and Larsson, 2014).

The TTR Bayes classifier assumes that the evidence is independent, i.e. that the probability of each piece of evidence is independent of every other piece of evidence. This is a severe limitation, but we adopt it here only as a simplifying assumption. Our Probabilistic TTR framework is by no means limited to learning based on such independence assumptions, and we will discard them when we move into fully articulated Bayesian networks in future work.

We first formulate Bayes' rule of conditional probability. This defines the conditional probability of a conclusion $r : T_c$, given evidence $r : T_{e_1}, r : T_{e_2}, \ldots, r : T_{e_n}$, in terms of conditional probabilities of the form $\mathrm{p}(s_i : T_{e_i} \mid s_i : T_c)$, $1 \leq i \leq n$, and *priors* for conclusion and evidence.

$$\mathrm{p}_{A,\mathfrak{I}}(r : T_c \mid r : T_{e_1}, \ldots, r : T_{e_n}) =$$
$$\mathrm{prior}_{\mathfrak{I}}(T_c)\frac{p_{A,\mathfrak{I}}(s : T_{e_1} \mid s : T_c) \ldots p_{A,\mathfrak{I}}(s : T_{e_n} \mid s : T_c)}{\mathrm{prior}_{\mathfrak{I}}(T_{e_1}) \ldots \mathrm{prior}_{\mathfrak{I}}(T_{e_n})}$$

The conditional probabilities, as well as the priors, will be computed from observations as explained earlier in this section. Part of the point of the rule of conditional probability is that it allows weighing several pieces of evidence together without requiring any previous observation of a situation involving all the evidence types. This allows classifying situations as being of types not previously encountered.

In cases where situation types are derived from natural language utterances, this allows us to deal with the compositional generativity of language, which allows that novel sentences corresponding to new situation types can easily be constructed. For instance, assume I have never classified any situation as being of the type where a boy smiles. In this case, if I have previously classified a situation as involving a boy, and another situation as involving someone (not a boy) smiling, I will still be able to assign a probability to a judgement that a situation is of a type specifying that a boy smiles. This will give me a probability that the sentence “A boy smiles” holds of the situation in question. In future work we will explore alternatives to the simple Bayesian approach to weighing together evidence from different sources used here, e.g. Dempster-Schafer theory (Shafer et al., 1976).

We also want the *posterior* probability of the judgement above (the probability of the judgement in light of the evidence). We obtain the posterior probabilities of the different possible conclusions by factoring in the probabilities of the evidence.

$$p_{A,\mathfrak{I}}(r : T_c) = p_{A,\mathfrak{I}}(s : T_c \mid s : T_{e_1}, \ldots, s : T_{e_n}) p_{A,\mathfrak{I}}(r : T_{e_1}) \ldots p_{A,\mathfrak{I}}(r : T_{e_n})$$

We now define a TTR Bayes classifier κ as a function from a situation s to a set of probabilistic Austinian propositions, defining a probability distribution over the possible conclusion types $T^{\kappa}_{c_1}, \ldots T^{\kappa}_{c_m}$, given a probability distribution over the set of evidence types $T^{\kappa}_{e_1}, \ldots T^{\kappa}_{e_n}$. If the classifier is a function

$$\kappa : \mathrm{Sit} \rightarrow \mathrm{Set}(\begin{bmatrix} \text{sit} & : & \mathit{Sit} \\ \text{sit-type} & : & \mathit{Type} \\ \text{prob} & : & [0,1] \end{bmatrix})$$

such that if r:*Sit*, then

$$\kappa(r) = \{ \begin{bmatrix} \text{sit} & = & r \\ \text{sit-type} & = & T \\ \text{prob} & = & \mathrm{p}^{\kappa}_{A,\mathfrak{J}}(r : T) \end{bmatrix} \mid T \in \langle \mathrm{T}^{\kappa}_{c_1}, \ldots, \mathrm{T}^{\kappa}_{c_m} \rangle \}$$

where

$$\mathrm{p}^{\kappa}_{A,\mathfrak{J}}(r : T) = \mathrm{p}_{A,\mathfrak{J}}(s : T \mid s : \mathrm{T}^{\kappa}_{e_1}, \ldots, s : \mathrm{T}^{\kappa}_{e_n})\mathrm{p}_{A,\mathfrak{J}}(r : T^{\kappa}_{e_1}) \ldots$$
$$\mathrm{p}_{A,\mathfrak{J}}(r : T^{\kappa}_{e_n})$$

A appends this set to $\mathfrak{J}$ as a result of observing and classifying a situation. The probabilities are then available for subsequent probabilistic reasoning.

4.2 Deriving a probabilistic model from Austinian propositions

One of the desiderata on our theory of learning is that we should be able to learn a probabilistic model specifying the probabilities of basic types (such as *Ind*) and ptypes. So far, we have only explained how an agent could learn probabilistic Austinian propositions such as $\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & \mathrm{T}_{\mathit{Pred}} \\ \text{prob} & = & p \end{bmatrix}$ where *Pred* is an n-place predicate and

$$\mathrm{T}_{\mathit{Pred}} = \begin{bmatrix} x_1 & : & \mathit{Ind} \\ \ldots & & \\ x_n & : & \mathit{Ind} \\ c_{\mathit{Pred}} & : & \mathit{Pred}(x_1, \ldots, x_n) \end{bmatrix}.$$

There seem to be good reasons for assuming that the probability that something is of type *Ind* is always 1. Something being of type *Ind* or not is a matter of definition rather than of judgement. It is only given the assumption that something is an individual that we can ascribe properties to it with some probability.

If we assume that judgements that something of type *Ind* are always 1, the probability of a judgement $s : \mathrm{T}_{\mathit{Pred}}$ is identical to the probability of the judgement $s.\mathrm{c}_{\mathit{Pred}} : \mathit{Pred}(s.x_1, \ldots, s.x_n)$. Hence, we can derive probabilities of ptype judgements from Austinian propositions of the $\mathrm{T}_{\mathit{Pred}}$ kind. For example, given an Austinian proposition

$$\left[\begin{array}{lll} \text{sit} & = & \left[\begin{array}{lll} \text{x} & = & \text{a}_{123} \\ \text{c}_{\text{smile}} & = & \text{e}_{456} \end{array}\right] \\ \text{sit-type} & = & \left[\begin{array}{lll} \text{x} & : & \textit{Ind} \\ \text{c}_{\text{smile}} & : & \text{smile(x)} \end{array}\right] \\ \text{prob} & = & 0.789 \end{array}\right],$$

we can derive the following model:
p(a_{123} : *Ind*) = 1.0
p(e_{456} : smile(a_{123})) = 0.789

In the general case, however, the string of observations $\mathfrak{I}$ may contain several probabilistic Austinian propositions concerning the same situation s and situation type T:

$$\mathfrak{I} = \{\ldots, \left[\begin{array}{lll} \text{sit} & = & s \\ \text{sit-type} & = & T \\ \text{prob} & = & p_1 \end{array}\right], \ldots, \left[\begin{array}{lll} \text{sit} & = & s \\ \text{sit-type} & = & T \\ \text{prob} & = & p_2 \end{array}\right], \ldots,$$

$$\left[\begin{array}{lll} \text{sit} & = & s \\ \text{sit-type} & = & T \\ \text{prob} & = & p_n \end{array}\right], \ldots\}$$

This may happen, for example, if s is classified from several perspectives as being of type T, or if information from several different sensors classify s as being of type T. For such cases, we need a way of combining several probabilities of $s : T$, delivered by several sources of information, into a single probability. One method is to take the maximum probability, while another is to use the mean probability or a weighted mean. The theory of probabilistic TTR is not required to decide this issue.

4.3 Discussion

We have made the simplifying assumption here that agents know what the relevant evidence is for each classifier. But in general it is not the case that an agent knows which features are relevant for determining that an individual is of a certain type. An agent could initially employ all available sensors with which it can observe a situation, and build an individual classifier for each kind of constraint. Regression analysis can be used to figure out which features are relevant for a certain classifier, and a threshold can be applied to rule out those features whose contribution is negligible to the classification task. Adding multiple features in classification (especially those that do not contribute to decisions about the class) increases the complexity of the learning task. Many more learning instances will be required to avoid acquiring spurious relations. Knowing which sensory inputs (or features) are useful contributors to a class is a component of our world knowledge which can be acquired through linguistic interaction with other agents (see for example Cooper and Larsson (2009)). In this way, world knowledge directs and simplifies perceptual classification.

In the current proposal, probabilities are computed from probabilistic judgements when needed – a kind of "lazy evaluation". Alternatively, we could pursue an incremental approach where probabilities are updated after each judgement, instead of recomputing them when they are needed. A further possibility is to combine lazy evaluation with incrementality, so that all and only the probabilities which have once been lazily computed are incrementally updated.

5 Conclusions and Future Work

We have presented a probabilistic version of TTR, relying heavily on classical probability equations for types formed with meet, join, and negation. This has permitted us to sustain classical equivalences and Boolean negation for complex types within an intensional type theory. We have replaced the truth of a type judgement with the probability of its being the case, and we have applied this approach both to judgements that a situation if of type T, and to the assertion that T is a non-empty type.

Our probabilistic formulation of TTR provides the basis for a compositional semantics in which functions apply to categorical semantic objects in order to return either functions from categorical interpretations to probabilistic functions, or, for sentences, to probabilistic Austinian propositions. One of the interesting ways in which this framework differs from classical model theoretic semantics is that the basic types and type judgements at the foundation of the type system correspond to perceptual judgements concerning objects and events in the world, rather than to entities in a model and set theoretic constructions defined on them.

We have offered a schematic view of semantic learning. On this account observations of situations in the world, support the acquisition of elementary Bayesian classifiers from which the basic probabilistic types of a TTR semantics are extracted. Our type theory is, then, the interface between observation-based learning of classifiers for objects and the situations in which they figure on one hand, and the computation of complex semantic values for the expressions of a natural language from these simple probabilistic types and type judgements on the other. Therefore our general model of interpretation achieves a highly integrated bottom-up treatment of linguistic meaning and perceptually-based cognition that situates meaning in learning how to make observational judgements concerning the likelihood of situations obtaining in the world.

Our future work will need to address both sides of the TTR interface. It is necessary to extend the coverage of our compositional fragment to deal with a fuller range of syntactic structures and their semantic properties at the high end of this model. At the lower end we must refine our learning theory to accommodate more complex kinds of classifier acquisition, and show how these classifiers feed the probabilistic type system that supports our compositional semantics. We must also extend the theory to incorporate dialogue and teacher driven learning.

One way of approaching the second of these tasks, which we are currently exploring, is to construct a robotic agent that learns functional types which apply to sensory data of a particular kind (records of situations) to return probabilistic predicate types. To the extent that this project is successful, it will demonstrate the viability of the model of learning and interpretation that we are proposing here.

Acknowledgments

Shalom Lappin's participation in the research reported here was funded by grant ES/J022969/1 from the Economic and Social Research Council of the UK, and a grant from the Wenner-Gren Foundations. We also gratefully acknowledge the support of Vetenskapsrådet, project 2009-1569, Semantic analysis of interaction and coordination in dialogue (SAICD); the Department of Philosophy, Linguistics, and Theory of Science; and the Centre for Language Technology at the University of Gothenburg. Some of the ideas in this paper were presented in a talk at the International Workshop on Type Theory with Records at the University of Gothenburg in December 2012, in a series of lectures at Labex, University of Paris 7 in April 2013, and at the EACL 2014 Workshop on Type Theory and Natural Language Semantics, Gothenburg, April 2014 (Cooper et al., 2014). We thank the audiences of these events for useful feedback. We are grateful to Alex Clark and Noah Goodman for invaluable comments on an earlier draft of this paper. We also thank Jekaterina Denissova, Jonathan Ginzburg, Michiel van Lambalgen, Dan Lassiter, Poppy Mankowitz, Aarne Ranta, and Peter Sutton for very helpful discussion of several of the proposals made here. We are grateful to three anonymous reviewers for their useful comments on an earlier draft of this paper. Of course we bear sole responsibility for any errors that may remain.

References

Austin, J. L. 1961. Truth. In J. O. Urmson and G. J. Warnock, eds., *J. L. Austin: Philosophical Papers*. Oxford: Oxford University Press.

Barwise, Jon and John Perry. 1983. *Situations and Attitudes*. Bradford Books. Cambridge, Mass.: MIT Press.

Carnap, R. 1947. *Meaning and Necessity*. Chicago: University of Chicago Press.

Chater, N. and C.D. Manning. 2006. Probabilistic models of language processing and acquisition. *Trends in Cognitive Sciences* 10(7):335–344.

Chater, N. and P. Vitányi. 2007. 'Ideal learning' of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology* 51(3):135–163.

Church, Alonzo. 1940. A formulation of the simple theory of types. *Journal of Symbolic Logic* 5(1):56–68.

Clark, Alexander. 2007. Learning deterministic context free grammars: the Omphalos competition. *Machine Learning* 66(1):93–110.

Clark, A. and S. Lappin. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Chichester, West Sussex, and Malden, MA: Wiley-Blackwell.

Cooper, Robin. 2005a. Austinian truth, attitudes and type theory. *Research on Language and Computation* 3:333–362.

Cooper, Robin. 2005b. Records and record types in semantic theory. *Journal of Logic and Computation* 15(2):99–112.

Cooper, Robin. 2012. Type theory and semantics in flux. In R. Kempson, N. Asher, and T. Fernando, eds., *Handbook of the Philosophy of Science*, vol. 14: Philosophy of Linguistics. Elsevier BV. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Cooper, Robin, Simon Dobnik, Shalom Lappin, and Staffan Larsson. 2014. A probabilistic rich type theory for semantic interpretation. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 72–79. Gothenburg, Association of Computational Linguistics.

Cooper, Robin and Jonathan Ginzburg. 2011. Negation in dialogue. In R. Artstein, M. Core, D. DeVault, K. Georgila, E. Kaiser, and A. Stent, eds., *SemDial 2011 (Los Angelogue): Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue*, pages 130–139.

Cooper, Robin and Jonathan Ginzburg. 2012. Negative inquistiveness and alternatives-based negation. In M. Aloni, V. Kimmelman, F. Roelofsen, G. W. Sassoon, K. Schulz, and M. Westera, eds., *Logic, Language and Meaning: 18th Amsterdam Colloquium, Amsterdam, The Netherlands, December 19–21, 2011, Revised Selected Papers*, no. 7218 in Lecture Notes in Computer Science, pages 32–41. Springer.

Cooper, Robin and Staffan Larsson. 2009. Compositional and ontological semantics in learning from corrective feedback and explicit definition. In

J. Edlund, J. Gustafson, A. Hjalmarsson, and G. Skantze, eds., *Proceedings of DiaHolmia: 2009 Workshop on the Semantics and Pragmatics of Dialogue*, pages 59–66. Department of Speech, Music and Hearing, KTH.

Dissanayake, M. W. M. G, P. M. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. 2001. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotic and Automation* 17(3):229–241.

Dobnik, Simon, Robin Cooper, and Staffan Larsson. 2013. Modelling language, action, and perception in Type Theory with Records. In D. Duchier and Y. Parmentier, eds., *Constraint Solving and Language Processing - 7th International Workshop on Constraint Solving and Language Processing, CSLP 2012, Orleans, France, September 13-14, 2012. Revised Selected Papers*, no. 8114 in Publications on Logic, Language and Information (FoLLI). Berlin, Heidelberg: Springer.

Dobnik, Simon, Robin Cooper, and Staffan Larsson. 2014. Type Theory with Records: a general framework for modelling spatial language. In S. Dobnik, R. Cooper, and S. Larsson, eds., *Proceedings of The Second Workshop on Action, Perception and Language (APL'2)*. The Fifth Swedish Language Technology Conference (SLTC), Uppsala, Sweden.

Fagin, Ronald, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. 1995. *Reasoning about knowledge*. Cambridge, Mass.: MIT Press. ISBN 0262061627.

Fernández, Raquel and Staffan Larsson. 2014. Vagueness and learning: A type-theoretic approach. In *Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (∗SEM 2014)*.

Fox, C. and S. Lappin. 2005. *Foundations of Intensional Semantics*. Oxford: Blackwell.

Fox, C. and S. Lappin. 2010. Expressiveness and complexity in underspecified semantics. *Linguistic Analysis, Festschrift for Joachim Lambek* 36:385–417.

Gaifman, H. and M. Snir. 1982. Probabilities over rich languages, randomness and testing. *The Journal of Symbolic Logic* 47(3):495–548.

Ginzburg, Jonathan. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford: Oxford University Press.

Goodman, N. and D. Lassiter. 2015. Probabilistic semantics and pragmatics: Uncertainty in language and thought. In S. Lappin and C. Fox, eds., *The Handbook of Contemporary Semantic Theory, Second Edition*. Oxford and Malden: Wiley-Blackwell.

Halpern, J. 2003. *Reasoning About Uncertainty*. Cambridge MA: MIT Press.

Heim, I. 1990. E-type pronouns and donkey anaphora,. *Linguistics and Philosophy* 13:137–177.

Jonsson, B. and A. Tarski. 1951. Boolean algebras with operators. *American Journal of Mathematics* 73:891–939.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.

Kolmogorov, A.N. 1950. *Foundations of Probability*. New York: Chelsea Publishing.

Kratzer, A. 2014. Situations in natural language semantics. In *Stanford Encylopedia of Philosophy*. Stanford University.

Kripke, S. 1959. A completeness theorem in modal logic. *Journal of Symbolic Logic* 24:1–14.

Lappin, S. 2000. An intensional parametric semantics for vague quantifiers. *Linguistics and Philosophy* 23:599–620.

Lappin, S. and N. Francez. 1994. E-type pronouns, I-sums, and donkey anaphora. *Linguistics and Philosophy* 17:391–428.

Larsson, Staffan. 2013. Formal semantics for perceptual classification. *Journal of Logic and Computation* .

Larsson, Staffan and Robin Cooper. 2009. Towards a formal view of corrective feedback. In A. Alishahi, T. Poibeau, and A. Villavicencio, eds., *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 1–9. EACL.

Lewis, David. 1973. *Conterfactuals*. Harvard University Press.

Luo, Z. 2010. Type-theoretical semantics with coercive subtyping. In *Proceedings of SALT 20*, pages 38–56.

Luo, Zhaohui. 2011. Contextual Analysis of Word Meanings in Type-Theoretical Semantics. In S. Pogodalla and J.-P. Prost, eds., *Logical Aspects of Computational Linguistics: 6th International Conference, LACL 2011*, no. 6736 in Lecture Notes in Artificial Intelligence, pages 159–174. Springer.

Martin-Löf, Per. 1984. *Intuitionistic Type Theory*. Naples: Bibliopolis.

Matuszek, Cynthia, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012a. A joint model of language and perception for grounded attribute learning. In J. Langford and J. Pineau, eds., *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*. Edinburgh, Scotland.

Matuszek, Cynthia, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. 2012b. Learning to parse natural language commands to a robot control system. In *Proc. of the 13th International Symposium on Experimental Robotics (ISER)*.

Montague, Richard. 1974. *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press. ed. and with an introduction by Richmond H. Thomason.

Papadimitriou, C. 1995. *Computational Complexity*. Readin, MA: Addison-Wesley Publishing Co.

Paris, J. B. 2010. Guangzhou winter school notes on pure inductive logic. University of Manchester, `http://www.maths.manchester.ac.uk/~jeff/lecture-notes/Guangzhou.pdf`.

Partee, Barbara H. 1977. Possible World Semantics and Linguistic Theory. *The Monist* 60(3):303–326.

Pearl, J. 1990. Bayesian decision methods. In G. Shafer and J. Pearl, eds., *Readings in Uncertain Reasoning*, pages 345–352. Morgan Kaufmann.

Perfors, A., J. Tenenbaum, and T. Regier. 2011. The learnability of abstract syntactic principles. *Cognition* 118(3):306–338.

Pollard, Carl. 2008. Hyperintensions. *Journal of Logic and Computation* 18:257–282.

Ranta, Aarne. 1994. *Type-Theoretical Grammar*. Oxford: Clarendon Press.

Rescher, Nicholas. 1999. How many possible worlds are there? *Philosophy and Phenomenological Research* 59(2):pp. 403–420.

Shafer, Glenn et al. 1976. *A mathematical theory of evidence*, vol. 1. Princeton university press Princeton.

Tenenbaum, Joshua B., Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science* 331(6022):1279–1285.

van Eijck, J. and S. Lappin. 2012. Probabilistic semantics for natural language. In Z. Christoff, P. Galeazzi, N. Gierasimszuk, A. Marcoci, and S. Smets, eds., *Logic and Interactive Rationality (LIRA), Volume 2*, pages 17–35. University of Amsterdam: ILLC.