A Rich Probabilistic Type Theory for the Semantics of Natural Language*

Shalom Lappin
King's College London

*Joint work with Robin Cooper, Simon Dobnik, and Staffan Larsson, University of Gothenburg

PReLiM Workshop
Charles University, Prague

July 8, 2014

Outline

Classical Semantic Theories

- Classical semantic theories (Montague (1974)), as well as dynamic (Kamp and Reyle (1993)) and underspecified (Fox and Lappin (2010)) frameworks use categorical type systems.
- A type T identifies a set of possible denotations for expressions in T.
- The theory specifies combinatorial operations for deriving the denotation of an expression from the values of its constituents.
- These theories cannot represent the gradience of semantic properties that is pervasive in speakers' judgements concerning truth, predication, and meaning relations.

Classical Semantic Theories

- Classical semantic theories (Montague (1974)), as well as dynamic (Kamp and Reyle (1993)) and underspecified (Fox and Lappin (2010)) frameworks use categorical type systems.
- A type T identifies a set of possible denotations for expressions in T.
- The theory specifies combinatorial operations for deriving the denotation of an expression from the values of its constituents.
- These theories cannot represent the gradience of semantic properties that is pervasive in speakers' judgements concerning truth, predication, and meaning relations.

Classical Semantic Theories

- Classical semantic theories (Montague (1974)), as well as dynamic (Kamp and Reyle (1993)) and underspecified (Fox and Lappin (2010)) frameworks use categorical type systems.
- A type T identifies a set of possible denotations for expressions in T.
- The theory specifies combinatorial operations for deriving the denotation of an expression from the values of its constituents.
- These theories cannot represent the gradience of semantic properties that is pervasive in speakers' judgements concerning truth, predication, and meaning relations.

Classical Semantic Theories

- Classical semantic theories (Montague (1974)), as well as dynamic (Kamp and Reyle (1993)) and underspecified (Fox and Lappin (2010)) frameworks use categorical type systems.
- A type *T* identifies a set of possible denotations for expressions in *T*.
- The theory specifies combinatorial operations for deriving the denotation of an expression from the values of its constituents.
- These theories cannot represent the gradience of semantic properties that is pervasive in speakers' judgements concerning truth, predication, and meaning relations.

Semantic Learning

- There is a fair amount of evidence indicating that language acquisition in general crucially relies on probabilistic learning (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).
- There is little (or no) psychological data to suggest that classical categorical type systems provide biologically determined constraints on semantic learning.

Semantic Learning

- There is a fair amount of evidence indicating that language acquisition in general crucially relies on probabilistic learning (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).
- There is little (or no) psychological data to suggest that classical categorical type systems provide biologically determined constraints on semantic learning.

Semantic Learning

- There is a fair amount of evidence indicating that language acquisition in general crucially relies on probabilistic learning (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).
- There is little (or no) psychological data to suggest that classical categorical type systems provide biologically determined constraints on semantic learning.

Semantic Learning

- There is a fair amount of evidence indicating that language acquisition in general crucially relies on probabilistic learning (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).
- There is little (or no) psychological data to suggest that classical categorical type systems provide biologically determined constraints on semantic learning.

Using Probability to Model Gradience and Learning

- A semantic theory that assigns probability rather than truth conditions to sentences is in a better position to deal with gradience and learning.
- Gradience is intrinsic to the theory by virtue of the fact that values are assigned to sentences in the continuum of real numbers [0,1], rather than Boolean values in {0,1}.
- A probabilistic account of semantic learning is facilitated if the target of learning is a probabilistic representation of meaning.
- Both semantic interpretation and semantic learning are characterised as reasoning under uncertainty.

Using Probability to Model Gradience and Learning

- A semantic theory that assigns probability rather than truth conditions to sentences is in a better position to deal with gradience and learning.
- Gradience is intrinsic to the theory by virtue of the fact that values are assigned to sentences in the continuum of real numbers [0,1], rather than Boolean values in {0,1}.
- A probabilistic account of semantic learning is facilitated if the target of learning is a probabilistic representation of meaning.
- Both semantic interpretation and semantic learning are characterised as reasoning under uncertainty.

Using Probability to Model Gradience and Learning

- A semantic theory that assigns probability rather than truth conditions to sentences is in a better position to deal with gradience and learning.
- Gradience is intrinsic to the theory by virtue of the fact that values are assigned to sentences in the continuum of real numbers [0,1], rather than Boolean values in {0,1}.
- A probabilistic account of semantic learning is facilitated if the target of learning is a probabilistic representation of meaning.
- Both semantic interpretation and semantic learning are characterised as reasoning under uncertainty.

Using Probability to Model Gradience and Learning

- A semantic theory that assigns probability rather than truth conditions to sentences is in a better position to deal with gradience and learning.
- Gradience is intrinsic to the theory by virtue of the fact that values are assigned to sentences in the continuum of real numbers [0,1], rather than Boolean values in {0,1}.
- A probabilistic account of semantic learning is facilitated if the target of learning is a probabilistic representation of meaning.
- Both semantic interpretation and semantic learning are characterised as reasoning under uncertainty.

Two Strategies

- On a top-down approach 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.
- On a bottom-up approach one defines a probabilistic type theory.
- The probability value of a sentence is the output of a function that encodes probabilistic semantic type judgements associated with its predicative syntactic constituents.

Two Strategies

- On a top-down approach 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.
- On a bottom-up approach one defines a probabilistic type theory.
- The probability value of a sentence is the output of a function that encodes probabilistic semantic type judgements associated with its predicative syntactic constituents.

Two Strategies

- On a top-down approach 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.
- On a bottom-up approach one defines a probabilistic type theory.
- The probability value of a sentence is the output of a function that encodes probabilistic semantic type judgements associated with its predicative syntactic constituents.

Two Strategies

- On a top-down approach 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.
- On a bottom-up approach one defines a probabilistic type theory.
- The probability value of a sentence is the output of a function that encodes probabilistic semantic type judgements associated with its predicative syntactic constituents.

A Top-Down Theory

- van Eijck and Lappin (2012) 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 $\sum_{w \in W} P(w) = 1$.
- An interpretation of a language L in a model $M = \langle D, W, P \rangle$ is given in terms of the standard notion $w \models \phi$:

$$[\![\phi]\!]^M := \sum_{w_i \in W \wedge w_i \models \phi} P(w_i)$$

A Top-Down Theory

- van Eijck and Lappin (2012) 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 $\sum_{w \in W} P(w) = 1$.
- An interpretation of a language L in a model $M = \langle D, W, P \rangle$ is given in terms of the standard notion $w \models \phi$:

$$[\![\phi]\!]^M := \sum_{w_i \in W \wedge w_i \models \phi} P(w_i)$$

A Top-Down Theory

- van Eijck and Lappin (2012) 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 $\sum_{w \in W} P(w) = 1$.
- An interpretation of a language L in a model $M = \langle D, W, P \rangle$ is given in terms of the standard notion $w \models \phi$:

$$[\![\phi]\!]^M := \sum_{w_i \in W \wedge w_i \models \phi} P(w_i)$$

The Probability Calculus

- 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.

The Probability Calculus

- 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.

The Probability Calculus

- 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.

Advantages of the Top-Down Approach

- This theory retains classical type and model theories to determine the value of a sentence in a world.
- Therefore, it uses well understood formal systems at both levels of representation.
- It applies a standard probability calculus for computing the probability value of a sentence.

Advantages of the Top-Down Approach

- This theory retains classical type and model theories to determine the value of a sentence in a world.
- Therefore, it uses well understood formal systems at both levels of representation.
- It applies a standard probability calculus for computing the probability value of a sentence.

Advantages of the Top-Down Approach

- This theory retains classical type and model theories to determine the value of a sentence in a world.
- Therefore, it uses well understood formal systems at both levels of representation.
- It applies a standard probability calculus for computing the probability value of a sentence.

Disadvantages of the Top-Down Approach

- It requires probabilities to be assigned to entire worlds in the model, with sentences receiving probability values derivatively from these assignments.
- Representing worlds (maximally consistent sets of propositions, or ultrafilters in a proof theoretic lattice of propositions) poses serious problems of tractability (Lappin (2014), Cooper et al. (2014)).
- The probability value of a sentence can only be computed relative to those of the other sentences of the language that specify the set of worlds (or possible situations).
- This holism seems to exclude the possibility of learning individual classifiers and type judgements independently of each other.

Disadvantages of the Top-Down Approach

- It requires probabilities to be assigned to entire worlds in the model, with sentences receiving probability values derivatively from these assignments.
- Representing worlds (maximally consistent sets of propositions, or ultrafilters in a proof theoretic lattice of propositions) poses serious problems of tractability (Lappin (2014), Cooper et al. (2014)).
- The probability value of a sentence can only be computed relative to those of the other sentences of the language that specify the set of worlds (or possible situations).
- This holism seems to exclude the possibility of learning individual classifiers and type judgements independently of each other.

Disadvantages of the Top-Down Approach

- It requires probabilities to be assigned to entire worlds in the model, with sentences receiving probability values derivatively from these assignments.
- Representing worlds (maximally consistent sets of propositions, or ultrafilters in a proof theoretic lattice of propositions) poses serious problems of tractability (Lappin (2014), Cooper et al. (2014)).
- The probability value of a sentence can only be computed relative to those of the other sentences of the language that specify the set of worlds (or possible situations).
- This holism seems to exclude the possibility of learning individual classifiers and type judgements independently of each other.

Disadvantages of the Top-Down Approach

- It requires probabilities to be assigned to entire worlds in the model, with sentences receiving probability values derivatively from these assignments.
- Representing worlds (maximally consistent sets of propositions, or ultrafilters in a proof theoretic lattice of propositions) poses serious problems of tractability (Lappin (2014), Cooper et al. (2014)).
- The probability value of a sentence can only be computed relative to those of the other sentences of the language that specify the set of worlds (or possible situations).
- This holism seems to exclude the possibility of learning individual classifiers and type judgements independently of each other.

A Bottom-Up Approach

- A bottom-up approach avoids the representability problem by assigning probabilities to individual type judgements as classifier applications.
- The probability of a sentence is determined relative to a bounded set of situation types, which can be learned as classifiers for situations.
- A bottom-up probabilistic semantics requires a probabilistic type theory.
- This theory provides the basis for an account of semantic learning in which situation type classifiers are acquired probabilistically through sampling and observation driven Bayesian inference.

A Bottom-Up Approach

- A bottom-up approach avoids the representability problem by assigning probabilities to individual type judgements as classifier applications.
- The probability of a sentence is determined relative to a bounded set of situation types, which can be learned as classifiers for situations.
- A bottom-up probabilistic semantics requires a probabilistic type theory.
- This theory provides the basis for an account of semantic learning in which situation type classifiers are acquired probabilistically through sampling and observation driven Bayesian inference.

A Bottom-Up Approach

- A bottom-up approach avoids the representability problem by assigning probabilities to individual type judgements as classifier applications.
- The probability of a sentence is determined relative to a bounded set of situation types, which can be learned as classifiers for situations.
- A bottom-up probabilistic semantics requires a probabilistic type theory.
- This theory provides the basis for an account of semantic learning in which situation type classifiers are acquired probabilistically through sampling and observation driven Bayesian inference.

A Bottom-Up Approach

- A bottom-up approach avoids the representability problem by assigning probabilities to individual type judgements as classifier applications.
- The probability of a sentence is determined relative to a bounded set of situation types, which can be learned as classifiers for situations.
- A bottom-up probabilistic semantics requires a probabilistic type theory.
- This theory provides the basis for an account of semantic learning in which situation type classifiers are acquired probabilistically through sampling and observation driven Bayesian inference.

Austinian Propositions

- We take probability to be distributed over situation types (Barwise and Perry (1983)).
- An Austinian proposition is a judgement that a situation is of a particular type, and we treat it as probabilistic.
- 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 is expressed probabilistically as $p(s : T) = r$, where $r \in [0,1]$.

Austinian Propositions

- We take probability to be distributed over situation types (Barwise and Perry (1983)).
- An Austinian proposition is a judgement that a situation is of a particular type, and we treat it as probabilistic.
- 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 is expressed probabilistically as $p(s : T) = r$, where $r \in [0,1]$.

Austinian Propositions

- We take probability to be distributed over situation types (Barwise and Perry (1983)).
- An Austinian proposition is a judgement that a situation is of a particular type, and we treat it as probabilistic.
- 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 is expressed probabilistically as $p(s : T) = r$, where $r \in [0,1]$.

Austinian Propositions

- We take probability to be distributed over situation types (Barwise and Perry (1983)).
- An Austinian proposition is a judgement that a situation is of a particular type, and we treat it as probabilistic.
- 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 is expressed probabilistically as $p(s : T) = r$, where $r \in [0,1]$.

Probabilistic TTR: Basic Types and PTypes

Our type system is based on Cooper's (2012) Type Theory with Records (TTR), and it includes the following types.

- **Basic Types** are not constructed out of other objects introduced in the theory.
  - If T is a basic type, $p(a : T)$ for any object a is provided by an assignment of probabilities to judgements involving basic types.
- **PTypes** are constructed from a *predicate* and an appropriate sequence of arguments.
  - *man(john,18:10)* is the type of situation where John is a man at time 18:10.
  - A probability model provides probabilities $p(e : r(a_1, \ldots, a_n))$ for ptypes $r(a_1, \ldots, a_n)$.
  - We take both common nouns and verbs to provide the components out of which PTypes are constructed.

Probabilistic TTR: Basic Types and PTypes

Our type system is based on Cooper's (2012) Type Theory with Records (TTR), and it includes the following types.

- **Basic Types** are not constructed out of other objects introduced in the theory.
  - If T is a basic type, $p(a : T)$ for any object a is provided by an assignment of probabilities to judgements involving basic types.
- **PTypes** are constructed from a *predicate* and an appropriate sequence of arguments.
  - *man(john,18:10)* is the type of situation where John is a man at time 18:10.
  - A probability model provides probabilities $p(e : r(a_1, \ldots, a_n))$ for ptypes $r(a_1, \ldots, a_n)$.
  - We take both common nouns and verbs to provide the components out of which PTypes are constructed.

Meets and Joins

- **Meets and Joins** give, for T_1 and T_2, the meet, $T_1 \wedge T_2$ and the join $T_1 \vee T_2$, respectively.
- $a : T_1 \wedge T_2$ 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).
- The probabilities for meet and join types are defined by the classical Kolmogorov (1950) equations.
    - $p(a : T_1 \wedge T_2) = p(a : T_1)p(a : T_2 \mid a : T_1)$ (equivalently, $p(a : T_1 \wedge T_2) = p(a : T_1, a : T_2)$)
    - $p(a : T_1 \vee T_2) = p(a : T_1) + p(a : T_2) - p(a : T_1 \wedge T_2)$

Meets and Joins

- **Meets and Joins** give, for T_1 and T_2, the meet, $T_1 \wedge T_2$ and the join $T_1 \vee T_2$, respectively.
- $a : T_1 \wedge T_2$ 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).
- The probabilities for meet and join types are defined by the classical Kolmogorov (1950) equations.
  - $p(a : T_1 \wedge T_2) = p(a : T_1)p(a : T_2 \mid a : T_1)$ (equivalently, $p(a : T_1 \wedge T_2) = p(a : T_1, a : T_2)$)
  - $p(a : T_1 \vee T_2) = p(a : T_1) + p(a : T_2) - p(a : T_1 \wedge T_2)$

Meets and Joins

- **Meets and Joins** give, for T_1 and T_2, the meet, $T_1 \wedge T_2$ and the join $T_1 \vee T_2$, respectively.
- $a : T_1 \wedge T_2$ 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).
- The probabilities for meet and join types are defined by the classical Kolmogorov (1950) equations.
  - $p(a : T_1 \wedge T_2) = p(a : T_1)p(a : T_2 \mid a : T_1)$ (equivalently, $p(a : T_1 \wedge T_2) = p(a : T_1, a : T_2)$)
  - $p(a : T_1 \vee T_2) = p(a : T_1) + p(a : T_2) - p(a : T_1 \wedge T_2)$

Meets and Joins

- **Meets and Joins** give, for T_1 and T_2, the meet, $T_1 \wedge T_2$ and the join $T_1 \vee T_2$, respectively.
- $a : T_1 \wedge T_2$ 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).
- The probabilities for meet and join types are defined by the classical Kolmogorov (1950) equations.
  - $p(a : T_1 \wedge T_2) = p(a : T_1)p(a : T_2 \mid a : T_1)$ (equivalently, $p(a : T_1 \wedge T_2) = p(a : T_1, a : T_2)$)
  - $p(a : T_1 \vee T_2) = p(a : T_1) + p(a : T_2) - p(a : T_1 \wedge T_2)$

Subtypes

- **Subtypes**: 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 $a : T_1$.
- 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 that $p(a : T_1 \wedge T_2) \leq p(a : T_1)$, and $p(a : T_1) \leq p(a : T_1 \vee T_2)$.

Subtypes

- **Subtypes**: 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 $a : T_1$.
- 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 that $p(a : T_1 \wedge T_2) \leq p(a : T_1)$, and $p(a : T_1) \leq p(a : T_1 \vee T_2)$.

Subtypes

- **Subtypes**: 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 $a : T_1$.
- 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 that $p(a : T_1 \wedge T_2) \leq p(a : T_1)$, and $p(a : T_1) \leq p(a : T_1 \vee T_2)$.

Subtypes

- **Subtypes**: 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 $a : T_1$.
- 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 that $p(a : T_1 \wedge T_2) \leq p(a : T_1)$, and $p(a : T_1) \leq p(a : T_1 \vee T_2)$.

Subtypes

- **Subtypes**: 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 $a : T_1$.
- 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 that $p(a : T_1 \wedge T_2) \leq p(a : T_1)$, and $p(a : T_1) \leq p(a : T_1 \vee T_2)$.

Subtypes

- **Subtypes**: 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 $a : T_1$.
- 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 that $p(a : T_1 \wedge T_2) \leq p(a : T_1)$, and $p(a : T_1) \leq p(a : T_1 \vee T_2)$.

Generalized Probabilistic Meet

- 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 $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n) = 1$.
- Universal quantification is an unbounded conjunctive probability, which is true if it is vacuously satisfied ($n = 0$) (Paris (2010)).
- **Conditional Conjunctive Probabilities**:

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

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

  If $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n \mid a : T) = 1$.

Generalized Probabilistic Meet

- 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 $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n) = 1$.
- Universal quantification is an unbounded conjunctive probability, which is true if it is vacuously satisfied ($n = 0$) (Paris (2010)).
- **Conditional Conjunctive Probabilities**:

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

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

  If $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n \mid a : T) = 1$.

Generalized Probabilistic Meet

- 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 $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n) = 1$.
- Universal quantification is an unbounded conjunctive probability, which is true if it is vacuously satisfied ($n = 0$) (Paris (2010)).
- **Conditional Conjunctive Probabilities**:

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

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

  If $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n \mid a : T) = 1$.

Generalized Probabilistic Meet

- 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 $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n) = 1$.
- Universal quantification is an unbounded conjunctive probability, which is true if it is vacuously satisfied ($n = 0$) (Paris (2010)).
- **Conditional Conjunctive Probabilities**:

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

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

  If $n = 0$, $\bigwedge_p(a_0 : T_0, \ldots, a_n : T_n \mid a : T) = 1$.

Generalized Probabilistic Meet

- 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 $n = 0$, $\bigwedge_p (a_0 : T_0, \ldots, a_n : T_n) = 1$.
- Universal quantification is an unbounded conjunctive probability, which is true if it is vacuously satisfied ($n = 0$) (Paris (2010)).
- **Conditional Conjunctive Probabilities**:

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

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

  If $n = 0$, $\bigwedge_p (a_0 : T_0, \ldots, a_n : T_n \mid a : T) = 1$.

Generalized Probabilistic Join

- Let $\bigvee^{p}(a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n)$ be the *disjunctive probability of judgements* $a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n$.
- $\bigvee^{p}(a_0 : T_0, \ldots, a_n : T_n) =$

  $\bigvee^{p}(a_0 : T_0, \ldots, a_{n-1} : T_{n-1}) + p(a_n : T_n) - \bigwedge_{p}(a_0 : T_0, \ldots, a_{n-1} : T_{n-1})p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1})$
- If $n = 0$, $\bigvee^{p}(a_0 : T_0, \ldots, a_n : T_n) = 0$.
- Existential quantification is an unbounded disjunctive probability, which is false if it lacks a single non-nil probability instance ($n = 0$).

Generalized Probabilistic Join

- Let $\bigvee^p(a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n)$ be the *disjunctive probability of judgements* $a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n$.
- $\bigvee^p(a_0 : T_0, \ldots, a_n : T_n) =$

  $\bigvee^p(a_0 : T_0, \ldots, a_{n-1} : T_{n-1}) + p(a_n : T_n) - \bigwedge_p(a_0 : T_0, \ldots, a_{n-1} : T_{n-1})p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1})$
- If $n = 0$, $\bigvee^p(a_0 : T_0, \ldots, a_n : T_n) = 0$.
- Existential quantification is an unbounded disjunctive probability, which is false if it lacks a single non-nil probability instance ($n = 0$).

Generalized Probabilistic Join

- Let $\bigvee^{p}(a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n)$ be the *disjunctive probability of judgements* $a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n$.
- $\bigvee^{p}(a_0 : T_0, \ldots, a_n : T_n) =$

  $\bigvee^{p}(a_0 : T_0, \ldots, a_{n-1} : T_{n-1}) + p(a_n : T_n) - \bigwedge_{p}(a_0 : T_0, \ldots, a_{n-1} : T_{n-1})p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1})$
- If $n = 0$, $\bigvee^{p}(a_0 : T_0, \ldots, a_n : T_n) = 0$.
- Existential quantification is an unbounded disjunctive probability, which is false if it lacks a single non-nil probability instance ($n = 0$).

Generalized Probabilistic Join

- Let $\bigvee^{p}(a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n)$ be the *disjunctive probability of judgements* $a_0 : T_0, a_1 : T_1, \ldots, a_n : T_n$.
- $\bigvee^{p}(a_0 : T_0, \ldots, a_n : T_n) =$

  $\bigvee^{p}(a_0 : T_0, \ldots, a_{n-1} : T_{n-1}) + p(a_n : T_n) - \bigwedge_{p}(a_0 : T_0, \ldots, a_{n-1} : T_{n-1}) p(a_n : T_n \mid a_0 : T_0, \ldots, a_{n-1} : T_{n-1})$
- If $n = 0$, $\bigvee^{p}(a_0 : T_0, \ldots, a_n : T_n) = 0$.
- Existential quantification is an unbounded disjunctive probability, which is false if it lacks a single non-nil probability instance ($n = 0$).

Function Types

- **Function Types** give, for any types T_1 and T_2, the type $(T_1 \to T_2)$.
- This 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.
- The probability that a function f is of type $(T_1 \to T_2)$ is the probability that everything in its domain is of type T_1, that everything in its range is of type T_2, and that everything not in its domain which has some probability of being of type T_1 is *not*, in fact, of type T_1
- $p(f : (T_1 \to T_2)) = \bigwedge^{p}_{a \in \mathrm{dom}(f)} (a : T_1, f(a) : T_2)(1 - \bigvee^{p}_{a \notin \mathrm{dom}(f)} (a : T_1))$

Function Types

- **Function Types** give, for any types T_1 and T_2, the type $(T_1 \rightarrow T_2)$.
- This 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.
- 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, that everything in its range is of type T_2, and that everything not in its domain which has some probability of being of type T_1 is *not*, in fact, of type T_1
- $p(f : (T_1 \rightarrow T_2)) = \bigwedge^{p}_{a \in \mathrm{dom}(f)} (a : T_1, f(a) : T_2)(1 - \bigvee^{p}_{a \notin \mathrm{dom}(f)} (a : T_1))$

Function Types

- **Function Types** give, for any types T_1 and T_2, the type $(T_1 \to T_2)$.
- This 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.
- The probability that a function f is of type $(T_1 \to T_2)$ is the probability that everything in its domain is of type T_1, that everything in its range is of type T_2, and that everything not in its domain which has some probability of being of type T_1 is *not*, in fact, of type T_1
- $p(f : (T_1 \to T_2)) = \bigwedge^{p}_{a \in \text{dom}(f)} (a : T_1, f(a) : T_2)(1 - \bigvee^{p}_{a \notin \text{dom}(f)} (a : T_1))$

Function Types

- **Function Types** give, for any types T_1 and T_2, the type $(T_1 \rightarrow T_2)$.
- This 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.
- 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, that everything in its range is of type T_2, and that everything not in its domain which has some probability of being of type T_1 is *not*, in fact, of type T_1
- $p(f : (T_1 \rightarrow T_2)) = \bigwedge^{p}_{a \in \mathrm{dom}(f)} (a : T_1, f(a) : T_2)(1 - \bigvee^{p}_{a \notin \mathrm{dom}(f)} (a : T_1))$

Function Types: Example 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.
- Let f map lightning events to thunder events, and and let f have as its domain all events which have been judged to have probability greater than 0 of being lightning events.
- Assume all putative lightning events are clear examples of lightning and are associated by f with clear events of thunder.
- If there are four such pairs of events, then the probability of f being of type $(T_1 \rightarrow T_2)$ is $(1 \times 1)^4 = 1$.

Function Types: Example 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.
- Let f map lightning events to thunder events, and and let f have as its domain all events which have been judged to have probability greater than 0 of being lightning events.
- Assume all putative lightning events are clear examples of lightning and are associated by f with clear events of thunder.
- If there are four such pairs of events, then the probability of f being of type $(T_1 \to T_2)$ is $(1 \times 1)^4 = 1$.

Function Types: Example 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.
- Let f map lightning events to thunder events, and and let f have as its domain all events which have been judged to have probability greater than 0 of being lightning events.
- Assume all putative lightning events are clear examples of lightning and are associated by f with clear events of thunder.
- If there are four such pairs of events, then the probability of f being of type $(T_1 \to T_2)$ is $(1 \times 1)^4 = 1$.

Function Types: Example 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.
- Let f map lightning events to thunder events, and and let f have as its domain all events which have been judged to have probability greater than 0 of being lightning events.
- Assume all putative lightning events are clear examples of lightning and are associated by f with clear events of thunder.
- If there are four such pairs of events, then the probability of f being of type $(T_1 \rightarrow T_2)$ is $(1 \times 1)^4 = 1$.

Function Types: Example 2

- Alternatively, suppose that for for of the four events f associates a lightning event with a silent event.
- 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 $(T_1 \rightarrow T_2)$.

Function Types: Example 2

- Alternatively, suppose that for for of the four events f associates a lightning event with a silent event.
- 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 $(T_1 \rightarrow T_2)$.

Function Types: Example 2

- Alternatively, suppose that for for of the four events f associates a lightning event with a silent event.
- 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 $(T_1 \rightarrow T_2)$.

Increasing the Size of the Domain of a Function Type

- If 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 $\mathrm{dom}(f)$.
- If, 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 a_i that is added to $\mathrm{dom}(f)$ 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.

Increasing the Size of the Domain of a Function Type

- If 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 $\mathrm{dom}(f)$.
- If, 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 a_i that is added to $\mathrm{dom}(f)$ 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.

Increasing the Size of the Domain of a Function Type

- If 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 $\mathrm{dom}(f)$.
- If, 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 a_i that is added to $\mathrm{dom}(f)$ 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.

Increasing the Size of the Domain of a Function Type

- If 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 $\mathrm{dom}(f)$.
- If, 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 a_i that is added to $\mathrm{dom}(f)$ 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.

Increasing the Size of the Domain of a Function Type

- If 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 $\mathrm{dom}(f)$.
- If, 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 a_i that is added to $\mathrm{dom}(f)$ 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.

Function Type Judgements as Universally Quantified Assertions

- We are interpreting a functional type judgement of this kind as a universally quantified assertion over the pairing of objects in $\mathrm{dom}(f)$ and $\mathrm{range}(f)$.
- 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.
- 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 Type Judgements as Universally Quantified Assertions

- We are interpreting a functional type judgement of this kind as a universally quantified assertion over the pairing of objects in $\mathrm{dom}(f)$ and $\mathrm{range}(f)$.
- 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.
- 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 Type Judgements as Universally Quantified Assertions

- We are interpreting a functional type judgement of this kind as a universally quantified assertion over the pairing of objects in $\mathrm{dom}(f)$ and $\mathrm{range}(f)$.
- 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.
- 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 Type Judgements as Universally Quantified Assertions

- We are interpreting a functional type judgement of this kind as a universally quantified assertion over the pairing of objects in $\mathrm{dom}(f)$ and $\mathrm{range}(f)$.
- 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.
- 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.

Negation and Instantiation of Types

- **Negation**: $\neg T$, of type T, is the function type $(T \rightarrow \bot)$, where $\bot$ is a necessarily empty type and $p(\bot) = 0$.
- It follows from our rules for function types that $p(f : \neg T) = 1$ if $\mathrm{dom}(f) = \emptyset$, ($T$ is empty, and 0 otherwise).
- We also assign probabilities to judgements concerning the (non-)emptiness of a type, $p(T)$.
- Our account of negation entails that $p(T \vee \neg T) = 1$, and (ii) $p(\neg\neg T) = p(T)$.
- Therefore, we sustain classical Boolean negation and disjunction, in contrast to Martin-Löf's (1984) intuitionistic type theory.

Negation and Instantiation of Types

- **Negation**: $\neg T$, of type T, is the function type $(T \rightarrow \bot)$, where $\bot$ is a necessarily empty type and $p(\bot) = 0$.
- It follows from our rules for function types that $p(f : \neg T) = 1$ if $\mathrm{dom}(f) = \emptyset$, ($T$ is empty, and 0 otherwise).
- We also assign probabilities to judgements concerning the (non-)emptiness of a type, $p(T)$.
- Our account of negation entails that $p(T \vee \neg T) = 1$, and (ii) $p(\neg\neg T) = p(T)$.
- Therefore, we sustain classical Boolean negation and disjunction, in contrast to Martin-Löf's (1984) intuitionistic type theory.

Negation and Instantiation of Types

- **Negation**: $\neg T$, of type T, is the function type $(T \rightarrow \bot)$, where $\bot$ is a necessarily empty type and $p(\bot) = 0$.
- It follows from our rules for function types that $p(f : \neg T) = 1$ if $\mathrm{dom}(f) = \emptyset$, ($T$ is empty, and 0 otherwise).
- We also assign probabilities to judgements concerning the (non-)emptiness of a type, $p(T)$.
- Our account of negation entails that $p(T \vee \neg T) = 1$, and (ii) $p(\neg\neg T) = p(T)$.
- Therefore, we sustain classical Boolean negation and disjunction, in contrast to Martin-Löf's (1984) intuitionistic type theory.

Negation and Instantiation of Types

- **Negation**: $\neg T$, of type T, is the function type $(T \rightarrow \bot)$, where $\bot$ is a necessarily empty type and $p(\bot) = 0$.
- It follows from our rules for function types that $p(f : \neg T) = 1$ if $\mathrm{dom}(f) = \emptyset$, ($T$ is empty, and 0 otherwise).
- We also assign probabilities to judgements concerning the (non-)emptiness of a type, $p(T)$.
- Our account of negation entails that $p(T \vee \neg T) = 1$, and (ii) $p(\neg\neg T) = p(T)$.
- Therefore, we sustain classical Boolean negation and disjunction, in contrast to Martin-Löf's (1984) intuitionistic type theory.

Negation and Instantiation of Types

- **Negation**: $\neg T$, of type T, is the function type $(T \rightarrow \bot)$, where $\bot$ is a necessarily empty type and $p(\bot) = 0$.
- It follows from our rules for function types that $p(f : \neg T) = 1$ if $\mathrm{dom}(f) = \emptyset$, ($T$ is empty, and 0 otherwise).
- We also assign probabilities to judgements concerning the (non-)emptiness of a type, $p(T)$.
- Our account of negation entails that $p(T \vee \neg T) = 1$, and (ii) $p(\neg\neg T) = p(T)$.
- Therefore, we sustain classical Boolean negation and disjunction, in contrast to Martin-Löf's (1984) intuitionistic type theory.

Dependent Types

- **Dependent Types** are functions from objects to types.
- Given appropriate arguments as functions they will return a type.
- Therefore, the account of probabilities associated with functions above applies to dependent types.

Dependent Types

- **Dependent Types** are functions from objects to types.
- Given appropriate arguments as functions they will return a type.
- Therefore, the account of probabilities associated with functions above applies to dependent types.

Dependent Types

- **Dependent Types** are functions from objects to types.
- Given appropriate arguments as functions they will return a type.
- Therefore, the account of probabilities associated with functions above applies to dependent types.

Record Types

- **Record Types** are sets of ordered pairs (*fields*) whose first member is a label and whose second member is an object of some type, possibly itself a record, where records are functional on labels (each label in a record can only occur once in the record's left projection).
- If T is a record type, ℓ is a label not occuring 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, 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 $r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n)$.

Record Types

- **Record Types** are sets of ordered pairs (*fields*) whose first member is a label and whose second member is an object of some type, possibly itself a record, where records are functional on labels (each label in a record can only occur once in the record's left projection).
- If T is a record type, ℓ is a label not occuring 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, 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 $r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n)$.

Record Types

- **Record Types** are sets of ordered pairs (*fields*) whose first member is a label and whose second member is an object of some type, possibly itself a record, where records are functional on labels (each label in a record can only occur once in the record's left projection).
- If T is a record type, ℓ is a label not occuring 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, 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 $r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n)$.

Record Types

The probability that an object r is of a record type T:

1. $p(r : Rec) = 1$ if r is a record, 0 otherwise
2. $p(r : T_1 \cup \{\langle \ell, T_2 \rangle\}) = \bigwedge_p (r : T_1, r.\ell : T_2)$
3. If $\mathcal{T} : (T_1 \rightarrow (\ldots \rightarrow (T_n \rightarrow Type)\ldots))$, then
   $p(r : T \cup \{\langle \ell, \langle \mathcal{T}, \langle \pi_1, \ldots, \pi_n \rangle\rangle\rangle\}) =$
   $\bigwedge_p (r : T, r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n) \mid r.\pi_1 : T_1, \ldots, r.\pi_n : T_n)$

Record Types

The probability that an object r is of a record type T:

1. $p(r : Rec) = 1$ if r is a record, 0 otherwise
2. $p(r : T_1 \cup \{\langle \ell, T_2 \rangle\}) = \bigwedge_p (r : T_1, r.\ell : T_2)$
3. If $\mathcal{T} : (T_1 \rightarrow (\ldots \rightarrow (T_n \rightarrow Type) \ldots))$, then $p(r : T \cup \{\langle \ell, \langle \mathcal{T}, \langle \pi_1, \ldots, \pi_n \rangle \rangle \rangle\}) =$ $\bigwedge_p (r : T, r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n) \mid r.\pi_1 : T_1, \ldots, r.\pi_n : T_n)$

Record Types

The probability that an object r is of a record type T:

1. $p(r : Rec) = 1$ if r is a record, 0 otherwise
2. $p(r : T_1 \cup \{\langle \ell, T_2 \rangle\}) = \bigwedge_p (r : T_1, r.\ell : T_2)$
3. If $\mathcal{T} : (T_1 \rightarrow (\ldots \rightarrow (T_n \rightarrow Type)\ldots))$, then
   $p(r : T \cup \{\langle \ell, \langle \mathcal{T}, \langle \pi_1, \ldots, \pi_n \rangle \rangle \rangle \}) =$
   $\bigwedge_p (r : T, r.\ell : \mathcal{T}(r.\pi_1, \ldots, r.\pi_n) \mid r.\pi_1 : T_1, \ldots, r.\pi_n : T_n)$

A Strategy for Computing Probability Values of Complex Expressions

- Montague (1974) determines the denotation of a complex expression by applying a function to an intensional argument (as in $[\![$ NP $]\!]([\![\ ^\wedge$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, $[\![\cdot]\!]_p$ yields functions from categorical values to probabilities.
- For sentences it returns the probability that the sentence is true.

A Strategy for Computing Probability Values of Complex Expressions

- 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, $[\![\cdot]\!]_p$ yields functions from categorical values to probabilities.
- For sentences it returns the probability that the sentence is true.

A Strategy for Computing Probability Values of Complex Expressions

- 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, $[\![\cdot]\!]_p$ yields functions from categorical values to probabilities.
- For sentences it returns the probability that the sentence is true.

A Strategy for Computing Probability Values of Complex Expressions

- 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, $[\![\cdot]\!]_p$ yields functions from categorical values to probabilities.
- For sentences it returns the probability that the sentence is true.

A Probabilistic Compositonal Semantics

$$[\![\, [_S \; S_1 \text{ and } S_2] \,]\!]_p = p(\begin{bmatrix} e_1 : [\![\, S_1 \,]\!] \\ e_2 : [\![\, S_2 \,]\!] \end{bmatrix})$$

$$[\![\, [_S \; S_1 \text{ or } S_2] \,]\!]_p = p(\begin{bmatrix} e : [\![\, S_1 \,]\!] \vee [\![\, S_2 \,]\!] \end{bmatrix})$$

$$[\![\, [_S \; \text{Neg } S] \,]\!]_p = [\![\, \text{Neg} \,]\!]_p([\![\, S \,]\!])$$

$$[\![\, [_S \; \text{NP VP}] \,]\!]_p = [\![\, \text{NP} \,]\!]_p([\![\, \text{VP} \,]\!])$$

$$[\![\, [_{NP} \; \text{Det N}] \,]\!]_p = [\![\, \text{Det} \,]\!]_p([\![\, \text{N} \,]\!])$$

$$[\![\, [_{NP} \; N_{prop}] \,]\!]_p = [\![\, N_{prop} \,]\!]_p$$

$$[\![\, [_{VP} \; V_t \text{ NP}] \,]\!]_p = [\![\, V_t \,]\!]_p([\![\, \text{NP} \,]\!])$$

$$[\![\, [_{VP} \; V_i] \,]\!]_p = [\![\, V_i \,]\!]_p$$

A Probabilistic Compositonal Semantics

$[\![\ [_{\text{Neg}}\ \text{“it's not true that”}]\ ]\!]_p = \lambda T{:}RecType(\text{p}(\left[\text{e}{:}\neg T\right]))$

$[\![\ [_{\text{Det}}\ \text{“some”}]\ ]\!]_p = \lambda Q{:}Ppty(\lambda P{:}Ppty(\text{p}(\left[\text{e}{:}\text{some}(Q,\ P)\right])))$

$[\![\ [_{\text{Det}}\ \text{“every”}]\ ]\!]_p = \lambda Q{:}Ppty(\lambda P{:}Ppty(\text{p}(\left[\text{e}{:}\text{every}(Q,\ P)\right])))$

$[\![\ [_{\text{Det}}\ \text{“most”}]\ ]\!]_p = \lambda Q{:}Ppty(\lambda P{:}Ppty(\text{p}(\left[\text{e}{:}\text{most}(Q,\ P)\right])))$

$[\![\ [_{\text{N}}\ \text{“boy”}]\ ]\!]_p = \lambda r{:}\left[\text{x}{:}Ind\right](\text{p}(\left[\text{e}{:}\text{boy}(r.\text{x})\right]))$

$[\![\ [_{\text{N}}\ \text{“girl”}]\ ]\!]_p = \lambda r{:}\left[\text{x}{:}Ind\right](\text{p}(\left[\text{e}{:}\text{girl}(r.\text{x})\right]))$

$[\![\ [_{\text{Adj}}\ \text{“green”}]\ ]\!]_p = \lambda P{:}Ppty(\lambda r{:}\left[\text{x}{:}Ind\right](\text{p}((\left[\text{e}{:}\text{green}(r.\text{x},P)\right])))))$

$[\![\ [_{\text{Adj}}\ \text{“imaginary”}]\ ]\!]_p = \lambda P{:}Ppty(\lambda r{:}\left[\text{x}{:}Ind\right](\text{p}((\left[\text{e}{:}\text{imaginary}(r.\text{x},P)\right])))))$

A Probabilistic Compositonal Semantics

$$[\![\ [\mathrm{N}_{prop}\ \text{“Kim”}]\ ]\!]_p = \lambda P{:}Ppty(\mathrm{p}(P(\left[\mathrm{x{=}kim}\right])))$$

$$[\![\ [\mathrm{N}_{prop}\ \text{“Sandy”}]\ ]\!]_p = \lambda P{:}Ppty(\mathrm{p}(P(\left[\mathrm{x{=}sandy}\right])))$$

$$[\![\ [\mathrm{V}_t\ \text{“knows”}]\ ]\!]_p = \lambda \mathcal{P}{:}Quant(\lambda r_1{:}\left[\mathrm{x}{:}Ind\right](\mathrm{p}(\mathcal{P}(\lambda r_2{:}(\left[\mathrm{e{:}know}(r_1.\mathrm{x},r_2.\mathrm{x})\right]))))))$$

$$[\![\ [\mathrm{V}_t\ \text{“sees”}]\ ]\!]_p = \lambda \mathcal{P}{:}Quant(\lambda r_1{:}\left[\mathrm{x}{:}Ind\right](\mathrm{p}(\mathcal{P}(\lambda r_2{:}(\left[\mathrm{e{:}see}(r_1.\mathrm{x},r_2.\mathrm{x})\right]))))))$$

$$[\![\ [\mathrm{V}_i\ \text{“smiles”}]\ ]\!]_p = \lambda r{:}\left[\mathrm{x}{:}Ind\right](\mathrm{p}(\left[\mathrm{e{:}smile}(r.\mathrm{x})\right]))$$

$$[\![\ [\mathrm{V}_i\ \text{“laughs”}]\ ]\!]_p = \lambda r{:}\left[\mathrm{x}{:}Ind\right](\mathrm{p}(\left[\mathrm{e{:}laugh}(r.\mathrm{x})\right]))$$

A Probability Distribution for the Fragment

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

$p_d(a : Ind) = 1$ if a is kim or sandy

$p_d(s : T) \in [0, 1]$ 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

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

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

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

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

A Probability Distribution for the Fragment

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

$p_d(a : Ind) = 1$ if a is kim or sandy

$p_d(s : T) \in [0, 1]$ 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

$p_d(a : [^\tau P]) = p_d(P(\left[\text{x=}a\right]))$

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

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

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

Probabilistic GQ Judgements

- 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 $P \rightarrow Q$.
- 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.

Probabilistic GQ Judgements

- 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 $P \rightarrow Q$.
- 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.

Probabilistic GQ Judgements

- 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 $P \rightarrow Q$.
- 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.

An Example

$[\![\ [_S\ [_{NP}\ [_{N_{prop}}\ \text{Kim}]]\ [_{VP}\ [_{V_i}\ \text{smiles}]]]\ ]\!]_p =$

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

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

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

An Example

- Suppose that $p_d(s_1\text{:smile(kim)}) = .7$, $p_d(s_2\text{:smile(kim)}) = .3$, $p_d(s_3\text{:smile(kim)}) = .4$, and there are no other situations s_i such that $p_d(s_i\text{:smile(kim)}) > 0$.
- 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.
- $p_d(\text{smile(kim)})=$
  $\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}, s_3 : \text{smile(kim)}) =$

  $\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}) + .4 -$

  $.4\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}) =$

  $(.7 + .3 - .7 \times .3) + .4 - .4(.7 + .3 - .7 \times .3) =$

  $.874$

An Example

- Suppose that $p_d(s_1\text{:smile(kim)}) = .7$, $p_d(s_2\text{:smile(kim)}) = .3$, $p_d(s_3\text{:smile(kim)}) = .4$, and there are no other situations s_i such that $p_d(s_i\text{:smile(kim)}) > 0$.
- 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.
- $p_d(\text{smile(kim)}) =$
  $\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}, s_3 : \text{smile(kim)}) =$
  $\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}) + .4 -$
  $.4\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}) =$
  $(.7 + .3 - .7 \times .3) + .4 - .4(.7 + .3 - .7 \times .3) =$
  $.874$

An Example

- Suppose that $p_d(s_1$:smile(kim)) = .7, $p_d(s_2$:smile(kim)) = .3, $p_d(s_3$:smile(kim)) = .4, and there are no other situations s_i such that $p_d(s_i$:smile(kim)) > 0.
- Assume that these probabilities are independent of each other; that is, $p_d(s_3$:smile(kim)) = $p_d(s_3$:smile(kim) | s_1:smile(kim), s_2:smile(kim)), and so on.
- p_d(smile(kim))=

  $\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}, s_3 : \text{smile(kim)})$ =

  $\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)}) + .4 -$

  $.4\bigvee^{p}{}_{d}(s_1 : \text{smile(kim)}, s_2 : \text{smile(kim)})$ =

  $(.7 + .3 - .7 \times .3) + .4 - .4(.7 + .3 - .7 \times .3)$ =

  .874

An Example

- $[\![\alpha]\!]_{p_d}$ is the result of computing $[\![\alpha]\!]_p$ with respect to the probability distribution *d*.
- $p_d(\left[\text{e:smile(kim)}\right]) = .874$.
- Hence $[\![\ [_S\ [_{NP}\ [_{N_{prop}}\ \text{Kim}]]\ [_{VP}\ [_{V_i}\ \text{smiles}]]]\ ]\!]_{p_d} = .874$.

An Example

- $[\![\alpha]\!]_{p_d}$ is the result of computing $[\![\alpha]\!]_p$ with respect to the probability distribution d.
- $p_d(\left[\text{e:smile(kim)}\right]) = .874$.
- Hence $[\![[_S [_{NP} [_{N_{prop}} \text{Kim}]] [_{VP} [_{V_i} \text{smiles}]]]]\!]_{p_d} = .874$.

An Example

- $[\![\ \alpha\ ]\!]_{p_d}$ is the result of computing $[\![\ \alpha\ ]\!]_p$ with respect to the probability distribution d.
- $p_d(\left[\text{e:smile(kim)}\right]) = .874$.
- Hence $[\![\ [_{\text{S}}\ [_{\text{NP}}\ [_{\text{N}_{prop}}\ \text{Kim}]]\ [_{\text{VP}}\ [_{\text{V}_i}\ \text{smiles}]]]\ ]\!]_{p_d} = .874$.

Probabilistic Austinian Propositions

- *Probabilistic Austinian propositions* are records of type

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

- They assert that the probability that a situation *s* is of type *Type* with the value of *prob*.
- The definition of $[\![\cdot]\!]_p$ specifies a compositional procedure for generating an Austinian proposition (record) of this type from the meanings of the syntactic constituents of a sentence.

Probabilistic Austinian Propositions

- *Probabilistic Austinian propositions* are records of type

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

- They assert that the probability that a situation *s* is of type *Type* with the value of *prob*.
- The definition of $[\![\cdot]\!]_p$ specifies a compositional procedure for generating an Austinian proposition (record) of this type from the meanings of the syntactic constituents of a sentence.

Probabilistic Austinian Propositions

- *Probabilistic Austinian propositions* are records of type

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

- They assert that the probability that a situation *s* is of type *Type* with the value of *prob*.
- The definition of $[\![\cdot]\!]_p$ specifies a compositional procedure for generating an Austinian proposition (record) of this type from the meanings of the syntactic constituents of a sentence.

Observations as Type Judgements

- We assume that agents track observed situations and their types, modelled as probabilistic Austinian propositions.
- An observation of a red object might yield the following Austinian proposition for some a:*Ind*, s_1:red(a)

$$\left[\begin{array}{lll} \text{sit} & = & \left[\begin{array}{lll} \text{ref} & = & a \\ \text{c}_{\text{red}} & = & s_1 \end{array}\right] \\ \text{sit-type} & = & \left[\begin{array}{lll} \text{ref} & : & \mathit{Ind} \\ \text{c}_{\text{red}} & : & \text{red(ref)} \end{array}\right] \\ \text{prob} & = & 0.7 \end{array}\right]$$

Observations as Type Judgements

- We assume that agents track observed situations and their types, modelled as probabilistic Austinian propositions.
- An observation of a red object might yield the following Austinian proposition for some a:*Ind*, s_1:red(a)

$$\begin{bmatrix} \text{sit} & = & \begin{bmatrix} \text{ref} & = & a \\ \text{c}_{\text{red}} & = & s_1 \end{bmatrix} \\ \text{sit-type} & = & \begin{bmatrix} \text{ref} & : & \textit{Ind} \\ \text{c}_{\text{red}} & : & \text{red(ref)} \end{bmatrix} \\ \text{prob} & = & 0.7 \end{bmatrix}$$

Computing the Probability of a Type Judgement

- When an agent A encounters a new situation s and wants to know if it is of type T or not, he/she uses probabilistic reasoning to determine the value of $p_{A,\mathfrak{J}}(s : T)$.
- This denotes the probability that agent A assigns with respect to prior judgements $\mathfrak{J}$ to s being of type T.

Computing the Probability of a Type Judgement

- When an agent A encounters a new situation s and wants to know if it is of type T or not, he/she uses probabilistic reasoning to determine the value of $p_{A,\mathfrak{J}}(s : T)$.
- This denotes the probability that agent A assigns with respect to prior judgements $\mathfrak{J}$ to s being of type T.

Priors on Type Judgements

- An agent, A, makes judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$.
- For a type, T, $\mathfrak{J}_T$ represents that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$.
- If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions, then $\| T \|_{\mathfrak{J}}$ represents the sum of all probabilities associated with T in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}_T} j.prob$).
- $\mathcal{P}(\mathfrak{J})$ is the sum of all probabilities in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}} j.prob$).
- $\text{prior}_{\mathfrak{J}}(T)$ represents the prior probability that anything is of type T given $\mathfrak{J}$.
- $\text{prior}_{\mathfrak{J}}(T) = \frac{\|T\|_{\mathfrak{J}}}{\mathcal{P}(\mathfrak{J})}$ if $\mathcal{P}(\mathfrak{J}) > 0$, and 0 otherwise.

Priors on Type Judgements

- An agent, A, makes judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$.
- For a type, T, $\mathfrak{J}_T$ represents that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$.
- If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions, then $\| T \|_{\mathfrak{J}}$ represents the sum of all probabilities associated with T in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}_T} j.prob$).
- $\mathcal{P}(\mathfrak{J})$ is the sum of all probabilities in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}} j.prob$).
- $\text{prior}_{\mathfrak{J}}(T)$ represents the prior probability that anything is of type T given $\mathfrak{J}$.
- $\text{prior}_{\mathfrak{J}}(T) = \frac{\|T\|_{\mathfrak{J}}}{\mathcal{P}(\mathfrak{J})}$ if $\mathcal{P}(\mathfrak{J}) > 0$, and 0 otherwise.

Priors on Type Judgements

- An agent, A, makes judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$.
- For a type, T, $\mathfrak{J}_T$ represents that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$.
- If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions, then $\| T \|_{\mathfrak{J}}$ represents the sum of all probabilities associated with T in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}_T} j.prob$).
- $\mathcal{P}(\mathfrak{J})$ is the sum of all probabilities in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}} j.prob$).
- $\text{prior}_{\mathfrak{J}}(T)$ represents the prior probability that anything is of type T given $\mathfrak{J}$.
- $\text{prior}_{\mathfrak{J}}(T) = \frac{\|T\|_{\mathfrak{J}}}{\mathcal{P}(\mathfrak{J})}$ if $\mathcal{P}(\mathfrak{J}) > 0$, and 0 otherwise.

Priors on Type Judgements

- An agent, A, makes judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$.
- For a type, T, $\mathfrak{J}_T$ represents that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$.
- If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions, then $\| T \|_{\mathfrak{J}}$ represents the sum of all probabilities associated with T in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}_T} j.prob$).
- $\mathcal{P}(\mathfrak{J})$ is the sum of all probabilities in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}} j.prob$).
- $\text{prior}_{\mathfrak{J}}(T)$ represents the prior probability that anything is of type T given $\mathfrak{J}$.
- $\text{prior}_{\mathfrak{J}}(T) = \frac{\|T\|_{\mathfrak{J}}}{\mathcal{P}(\mathfrak{J})}$ if $\mathcal{P}(\mathfrak{J}) > 0$, and 0 otherwise.

Priors on Type Judgements

- An agent, A, makes judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$.
- For a type, T, $\mathfrak{J}_T$ represents that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$.
- If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions, then $\| T \|_{\mathfrak{J}}$ represents the sum of all probabilities associated with T in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}_T} j.prob$).
- $\mathcal{P}(\mathfrak{J})$ is the sum of all probabilities in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}} j.prob$).
- $\text{prior}_{\mathfrak{J}}(T)$ represents the prior probability that anything is of type T given $\mathfrak{J}$.
- $\text{prior}_{\mathfrak{J}}(T) = \frac{\|T\|_{\mathfrak{J}}}{\mathcal{P}(\mathfrak{J})}$ if $\mathcal{P}(\mathfrak{J}) > 0$, and 0 otherwise.

Priors on Type Judgements

- An agent, A, makes judgements based on a finite string of probabilistic Austinian propositions, $\mathfrak{J}$.
- For a type, T, $\mathfrak{J}_T$ represents that set of Austinian propositions j such that j.sit-type $\sqsubseteq T$.
- If T is a type and $\mathfrak{J}$ a finite string of probabilistic Austinian propositions, then $\| T \|_{\mathfrak{J}}$ represents the sum of all probabilities associated with T in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}_T} j.prob$).
- $\mathcal{P}(\mathfrak{J})$ is the sum of all probabilities in $\mathfrak{J}$ ($\sum_{j \in \mathfrak{J}} j.prob$).
- $\text{prior}_{\mathfrak{J}}(T)$ represents the prior probability that anything is of type T given $\mathfrak{J}$.
- $\text{prior}_{\mathfrak{J}}(T) = \frac{\|T\|_{\mathfrak{J}}}{\mathcal{P}(\mathfrak{J})}$ if $\mathcal{P}(\mathfrak{J}) > 0$, and 0 otherwise.

A Type Theoretic Bayesian Rule for Conditional Probability

- $p_{A,\mathfrak{J}}(s : T_1 \mid s : T_2)$ is 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.
- A computes these conditional probabilities with the equation

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

Otherwise, $p_{A,\mathfrak{J}}(s : T_1 \mid s : T_2) = 0$.

A Type Theoretic Bayesian Rule for Conditional Probability

- $p_{A,\mathfrak{J}}(s : T_1 \mid s : T_2)$ is 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.
- A computes these conditional probabilities with the equation

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

Otherwise, $p_{A,\mathfrak{J}}(s : T_1 \mid s : T_2) = 0$.

Learning a TTR Bayes Classifier from Evidence

- *A* classifies a new situation *s* based on the prior judgements $\mathfrak{J}$, and the evidence that *A* acquires about *s*.
- This evidence has the form

  $$p_{A,\mathfrak{J}}(s : T_{e_1}), \ldots, p_{A,\mathfrak{J}}(s : T_{e_n}),$$

  where $T_{e_1}, \ldots, T_{e_n}$ are the *evidence types*.
- The TTR Bayes classifier assumes that the evidence is independent, in that the probability of each piece of evidence is independent of every other piece of evidence.

Learning a TTR Bayes Classifier from Evidence

- *A* classifies a new situation *s* based on the prior judgements $\mathfrak{J}$, and the evidence that *A* acquires about *s*.
- This evidence has the form

  $$p_{A,\mathfrak{J}}(s : T_{e_1}), \ldots, p_{A,\mathfrak{J}}(s : T_{e_n}),$$

  where $T_{e_1}, \ldots, T_{e_n}$ are the *evidence types*.
- The TTR Bayes classifier assumes that the evidence is independent, in that the probability of each piece of evidence is independent of every other piece of evidence.

Learning a TTR Bayes Classifier from Evidence

- *A* classifies a new situation *s* based on the prior judgements $\mathfrak{J}$, and the evidence that *A* acquires about *s*.
- This evidence has the form

  $$p_{A,\mathfrak{J}}(s : T_{e_1}), \ldots, p_{A,\mathfrak{J}}(s : T_{e_n}),$$

  where $T_{e_1}, \ldots, T_{e_n}$ are the *evidence types*.
- The TTR Bayes classifier assumes that the evidence is independent, in that the probability of each piece of evidence is independent of every other piece of evidence.

Bayes' Rule

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

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

$$\text{prior}_{\mathfrak{J}}(T_c)\frac{p_{A,\mathfrak{J}}(s{:}T_{e_1}|s{:}T_c)\ldots p_{A,\mathfrak{J}}(s{:}T_{e_n}|s{:}T_c)}{\text{prior}_{\mathfrak{J}}(T_{e_1})+\ldots+\text{prior}_{\mathfrak{J}}(T_{e_n})} =$$

$$\text{prior}_{\mathfrak{J}}(T_c)\frac{\frac{\|T_{e_1}\wedge T_c\|_{\mathfrak{J}}}{\|T_c\|_{\mathfrak{J}}}\ldots\frac{\|T_{e_n}\wedge T_c\|_{\mathfrak{J}}}{\|T_c\|_{\mathfrak{J}}}}{\text{prior}_{\mathfrak{J}}(T_{e_1})+\ldots+\text{prior}_{\mathfrak{J}}(T_{e_n})}$$

Bayes' Rule

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

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

$$\text{prior}_{\mathfrak{I}}(T_c)\frac{p_{A,\mathfrak{I}}(s{:}T_{e_1}|s{:}T_c)\ldots p_{A,\mathfrak{I}}(s{:}T_{en}|s{:}T_c)}{\text{prior}_{\mathfrak{I}}(T_{e_1})+\ldots+\text{prior}_{\mathfrak{I}}(T_{en})} =$$

$$\text{prior}_{\mathfrak{I}}(T_c)\frac{\frac{\|T_{e_1} \wedge T_c\|_{\mathfrak{I}}}{\|T_c\|_{\mathfrak{I}}}\ldots\frac{\|T_{en} \wedge T_c\|_{\mathfrak{I}}}{\|T_c\|_{\mathfrak{I}}}}{\text{prior}_{\mathfrak{I}}(T_{e_1})+\ldots+\text{prior}_{\mathfrak{I}}(T_{en})}$$

Bayes' Rule

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

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

$$\text{prior}_{\mathfrak{I}}(T_c)\frac{p_{A,\mathfrak{I}}(s{:}T_{e_1}|s{:}T_c)\ldots p_{A,\mathfrak{I}}(s{:}T_{e_n}|s{:}T_c)}{\text{prior}_{\mathfrak{I}}(T_{e_1})+\ldots+\text{prior}_{\mathfrak{I}}(T_{e_n})} =$$

$$\text{prior}_{\mathfrak{I}}(T_c)\frac{\frac{\|T_{e_1} \wedge T_c\|_{\mathfrak{I}}}{\|T_c\|_{\mathfrak{I}}}\ldots\frac{\|T_{e_n} \wedge T_c\|_{\mathfrak{I}}}{\|T_c\|_{\mathfrak{I}}}}{\text{prior}_{\mathfrak{I}}(T_{e_1})+\ldots+\text{prior}_{\mathfrak{I}}(T_{e_n})}$$

The Posterior Probabilities of Conclusions

- 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.

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

The Posterior Probabilities of Conclusions

- 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.

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

Defining the Classifier Function

- Associated with the classifier is a collection of evidence types $T_{e_1}, T_{e_2}, \ldots, T_{e_n}$ and a collection of possible conclusion types $T_{c_1}, T_{c_2}, \ldots, T_{c_m}$
- We define a TTR Bayes classifier as a function from a situation *s* to a set of probabilistic Austinian propositions, defining a probability distribution over the possible conclusion types.

Defining the Classifier Function

- Associated with the classifier is a collection of evidence types $T_{e_1}, T_{e_2}, \ldots, T_{e_n}$ and a collection of possible conclusion types $T_{c_1}, T_{c_2}, \ldots, T_{c_m}$
- We define a TTR Bayes classifier as a function from a situation s to a set of probabilistic Austinian propositions, defining a probability distribution over the possible conclusion types.

Defining the Classifier Function

- The classifier function is specified as follows.

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

such that if s:Sit then

$$\kappa(s) = \{\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & T \\ \text{prob} & = & \text{p}_{A,\mathfrak{I}}(s : T \mid s : \text{T}_{e_1}, \ldots, s : \text{T}_{en}) \\ & & \quad \text{p}_{A,\mathfrak{I}}(s : T_{e_1}) \ldots \text{p}_{A,\mathfrak{I}}(s : T_{en}) \end{bmatrix} \mid T \in \langle \text{T}_{c_1}, \ldots, \text{T}_{c_m} \rangle\}$$

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

Defining the Classifier Function

- The classifier function is specified as follows.

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

such that if s:Sit then

$$\kappa(s) = \left\{\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & T \\ \text{prob} & = & \text{p}_{A,\mathfrak{I}}(s : T \mid s : \text{T}_{e_1}, \ldots, s : \text{T}_{en}) \\ & & \quad \text{p}_{A,\mathfrak{I}}(s : T_{e_1}) \ldots \text{p}_{A,\mathfrak{I}}(s : T_{en}) \end{bmatrix} \middle|\ T \in \langle \text{T}_{c_1}, \ldots, \text{T}_{c_m} \rangle \right\}$$

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

Defining the Classifier Function

- The classifier function is specified as follows.

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

such that if s:Sit then

$$\kappa(s) = \{\begin{bmatrix} \text{sit} & = & s \\ \text{sit-type} & = & T \\ \text{prob} & = & \text{p}_{A,\mathfrak{J}}(s : T \mid s : \text{T}_{e_1}, \ldots, s : \text{T}_{e_n}) \\ & & \quad \text{p}_{A,\mathfrak{J}}(s : T_{e_1}) \ldots \text{p}_{A,\mathfrak{J}}(s : T_{e_n}) \end{bmatrix} \mid T \in \langle \text{T}_{c_1}, \ldots, \text{T}_{c_m} \rangle\}$$

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

Bayes Classifiers and Bayesian Networks

- We are using a type theoretic variant of the standard Bayesian formula for conditional probabilities: $p(A \mid B) = \frac{|A\&B|}{|B|}$.
- Instead of counting categorical instances, we sum the probabilities of judgements, because our "training data" consists of probabilistic observational type judgements.
- By using an observer's previous type judgements as the prior for the rule that computes the probability of a new event being of a given type, we have, in effect, compressed information that properly belongs in a Bayesian network (Pearl (1990)) into our specification of a TTR Bayes classifier.
- Our classifier outputs a probability distribution over possible conclusions, rather than a categorical judgement for the conclusion with the highest probability

Bayes Classifiers and Bayesian Networks

- We are using a type theoretic variant of the standard Bayesian formula for conditional probabilities: $p(A \mid B) = \frac{|A\&B|}{|B|}$.
- Instead of counting categorical instances, we sum the probabilities of judgements, because our “training data” consists of probabilistic observational type judgements.
- By using an observer’s previous type judgements as the prior for the rule that computes the probability of a new event being of a given type, we have, in effect, compressed information that properly belongs in a Bayesian network (Pearl (1990)) into our specification of a TTR Bayes classifier.
- Our classifier outputs a probability distribution over possible conclusions, rather than a categorical judgement for the conclusion with the highest probability

Bayes Classifiers and Bayesian Networks

- We are using a type theoretic variant of the standard Bayesian formula for conditional probabilities: $p(A \mid B) = \frac{|A\&B|}{|B|}$.
- Instead of counting categorical instances, we sum the probabilities of judgements, because our "training data" consists of probabilistic observational type judgements.
- By using an observer's previous type judgements as the prior for the rule that computes the probability of a new event being of a given type, we have, in effect, compressed information that properly belongs in a Bayesian network (Pearl (1990)) into our specification of a TTR Bayes classifier.
- Our classifier outputs a probability distribution over possible conclusions, rather than a categorical judgement for the conclusion with the highest probability

Bayes Classifiers and Bayesian Networks

- We are using a type theoretic variant of the standard Bayesian formula for conditional probabilities: $p(A \mid B) = \frac{|A\&B|}{|B|}$.
- Instead of counting categorical instances, we sum the probabilities of judgements, because our "training data" consists of probabilistic observational type judgements.
- By using an observer's previous type judgements as the prior for the rule that computes the probability of a new event being of a given type, we have, in effect, compressed information that properly belongs in a Bayesian network (Pearl (1990)) into our specification of a TTR Bayes classifier.
- Our classifier outputs a probability distribution over possible conclusions, rather than a categorical judgement for the conclusion with the highest probability

Conclusions

- Our probabilistic formulation of a rich type theory with records provides the basis for a compositional semantics in which functions apply to categorical semantic objects in order to return functions from categorical interpretations to probabilistic judgements.
- For sentences, the rules generate probabilistic Austinian propositions.
- This framework differs from classical model theoretic semantics, *inter alia*, in 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.

Conclusions

- Our probabilistic formulation of a rich type theory with records provides the basis for a compositional semantics in which functions apply to categorical semantic objects in order to return functions from categorical interpretations to probabilistic judgements.
- For sentences, the rules generate probabilistic Austinian propositions.
- This framework differs from classical model theoretic semantics, *inter alia*, in 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.

Conclusions

- Our probabilistic formulation of a rich type theory with records provides the basis for a compositional semantics in which functions apply to categorical semantic objects in order to return functions from categorical interpretations to probabilistic judgements.
- For sentences, the rules generate probabilistic Austinian propositions.
- This framework differs from classical model theoretic semantics, *inter alia*, in 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.

Conclusions

- We have offered a schematic view of semantic learning in which observations of situations in the world support the acquisition of Bayes Classifiers.
- The basic probabilistic types of our type theoretical semantics are extracted from these classifiers.
- The proposed type theory specifies the interface between observation-based learning of classifiers for objects and situations, and the computation of complex semantic values for the expressions of a natural language.
- Our general model of interpretation achieves a highly integrated bottom-up treatment of linguistic meaning and perceptually-based cognition.
- It situates meaning in learning how to make observational judgements concerning the likelihood of situations obtaining in the world.

Conclusions

- We have offered a schematic view of semantic learning in which observations of situations in the world support the acquisition of Bayes Classifiers.
- The basic probabilistic types of our type theoretical semantics are extracted from these classifiers.
- The proposed type theory specifies the interface between observation-based learning of classifiers for objects and situations, and the computation of complex semantic values for the expressions of a natural language.
- Our general model of interpretation achieves a highly integrated bottom-up treatment of linguistic meaning and perceptually-based cognition.
- It situates meaning in learning how to make observational judgements concerning the likelihood of situations obtaining in the world.

Conclusions

- We have offered a schematic view of semantic learning in which observations of situations in the world support the acquisition of Bayes Classifiers.
- The basic probabilistic types of our type theoretical semantics are extracted from these classifiers.
- The proposed type theory specifies the interface between observation-based learning of classifiers for objects and situations, and the computation of complex semantic values for the expressions of a natural language.
- Our general model of interpretation achieves a highly integrated bottom-up treatment of linguistic meaning and perceptually-based cognition.
- It situates meaning in learning how to make observational judgements concerning the likelihood of situations obtaining in the world.

Conclusions

- We have offered a schematic view of semantic learning in which observations of situations in the world support the acquisition of Bayes Classifiers.
- The basic probabilistic types of our type theoretical semantics are extracted from these classifiers.
- The proposed type theory specifies the interface between observation-based learning of classifiers for objects and situations, and the computation of complex semantic values for the expressions of a natural language.
- Our general model of interpretation achieves a highly integrated bottom-up treatment of linguistic meaning and perceptually-based cognition.
- It situates meaning in learning how to make observational judgements concerning the likelihood of situations obtaining in the world.

Conclusions

- We have offered a schematic view of semantic learning in which observations of situations in the world support the acquisition of Bayes Classifiers.
- The basic probabilistic types of our type theoretical semantics are extracted from these classifiers.
- The proposed type theory specifies the interface between observation-based learning of classifiers for objects and situations, and the computation of complex semantic values for the expressions of a natural language.
- Our general model of interpretation achieves a highly integrated bottom-up treatment of linguistic meaning and perceptually-based cognition.
- It situates meaning in learning how to make observational judgements concerning the likelihood of situations obtaining in the world.

Future Work

- Bayesian reasoning from observation provides the incremental basis for learning and refining predicative types.
- In future work we will explore implementations of our learning theory in order to study the viability of our probabilistic type theory as an interface between perceptual judgement and compositional semantics.
- We hope to show that, in addition to its cognitive and theoretical interest, our proposed framework will yield results in robotic language learning, and dialogue modelling.

Future Work

- Bayesian reasoning from observation provides the incremental basis for learning and refining predicative types.
- In future work we will explore implementations of our learning theory in order to study the viability of our probabilistic type theory as an interface between perceptual judgement and compositional semantics.
- We hope to show that, in addition to its cognitive and theoretical interest, our proposed framework will yield results in robotic language learning, and dialogue modelling.

Future Work

- Bayesian reasoning from observation provides the incremental basis for learning and refining predicative types.
- In future work we will explore implementations of our learning theory in order to study the viability of our probabilistic type theory as an interface between perceptual judgement and compositional semantics.
- We hope to show that, in addition to its cognitive and theoretical interest, our proposed framework will yield results in robotic language learning, and dialogue modelling.