Knowledge Representation

So you know stuff... big whoop, I know stuff too, and even our computers know some stuff!

But knowledge is useless for communication if we're not speaking the same language... and if our computer knows stuff, we want to know it too!

> **Knowledge representation** attempts to coerce facts about the world into first order logic so that we can perform automated reasoning on it to discover new facts.

Because after all, isn't that the whole purpose of computers? To munch away at data and spit back some interesting findings?!</controversialComments></unmatchedTags>

> **Example**
> Some common uses of knowledge representation systems: medical databases linking diseases with symptoms, epidemics, etc.; drug interactions and pharmacological research; NLP and semantic webs.

So, naturally, we want the information that our computer spits back to be... well... natural, and readable to a human!

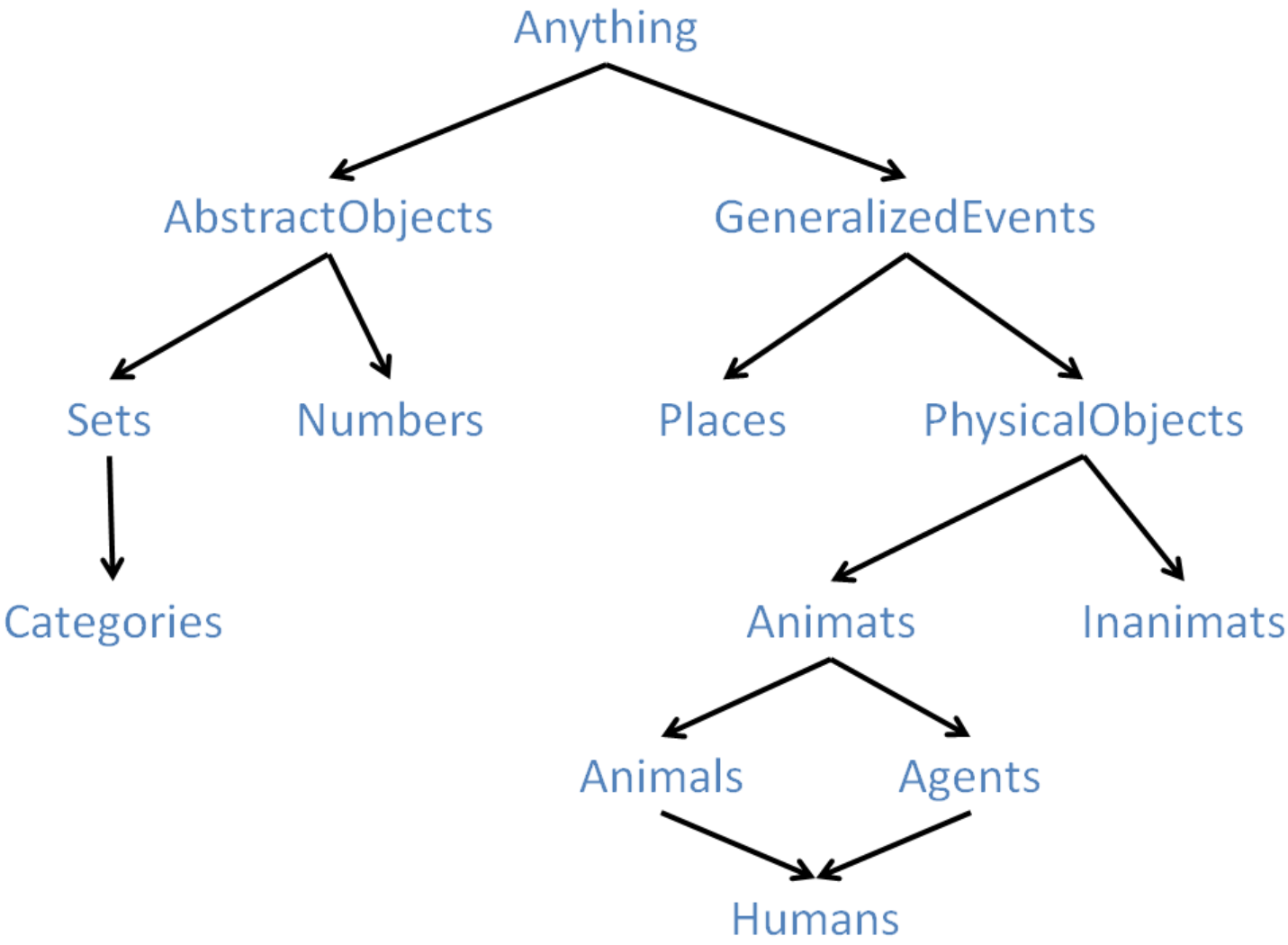
Before we do that, though, we want to organize our knowledge into an objective, unified representation such that communication is seemless and cross-domain integration is possible.

> An **ontology** is a classification system (usually organized into Knowledge Bases) such that we establish membership of specific world entities into classes or categories.

> An **upper ontology** is an ontology for general concepts that provides rankings of more specific sub-ontologies.

Here's a sample **upper ontology** for general classification, where parents of nodes are more general than their children.

NB, a node can have multiple parents in the case where it is correctly classified under multiple generalizations.



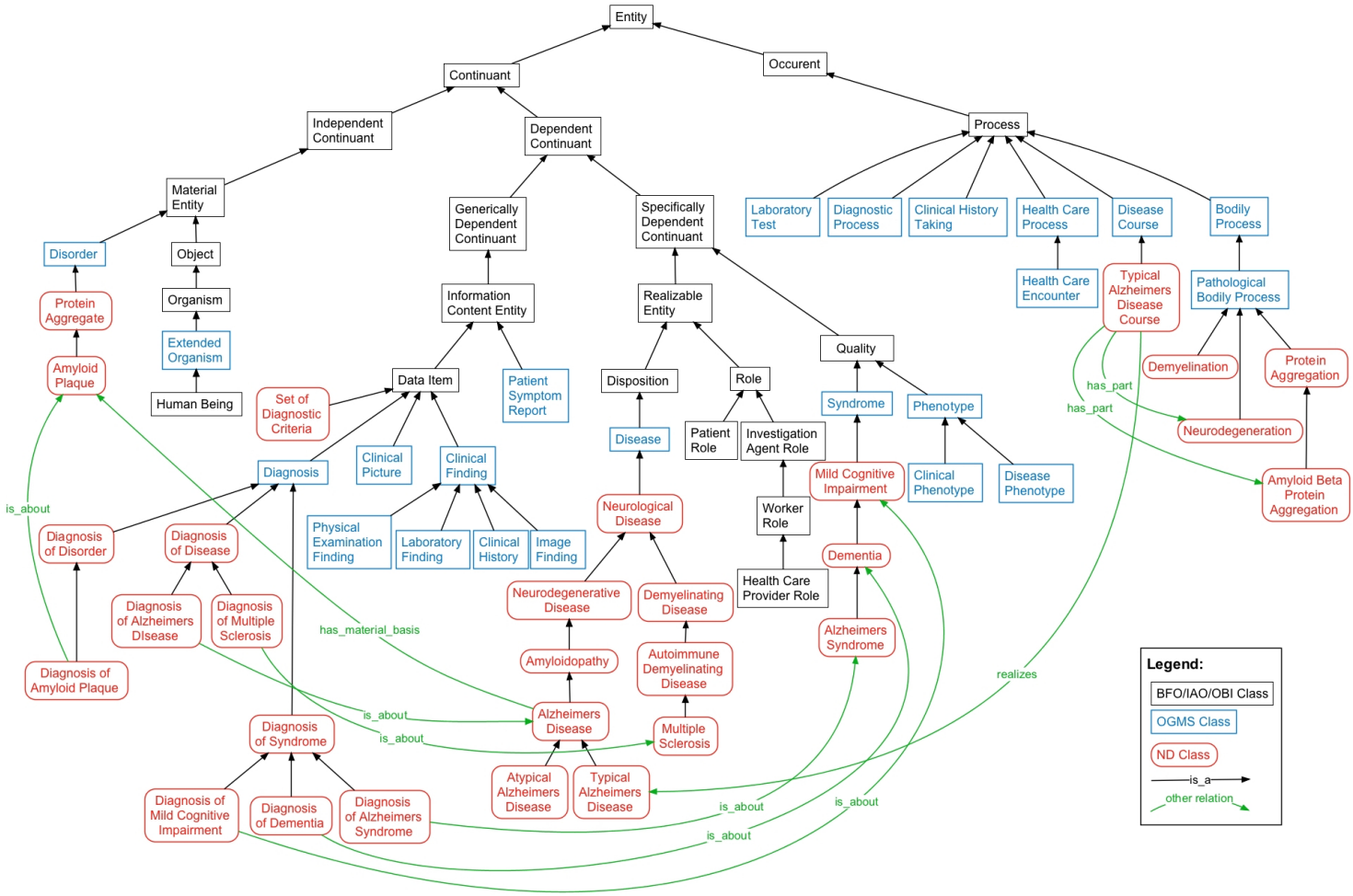
In practice, as you might've expected, creating and then deciding upon a unified ontology has been empirically difficult.

There have been a variety of efforts to produce unified ontologies, including:

- CYC (http://www.cyc.com/) develops semantic technologies for inference and data alignment.
- Openmind (http://openmind.media.mit.edu/) was an MIT project to crowd-source commonsense knowledge, which resulted in ConceptNet, useful for NLP.
- OWL (Web Ontology Language) (http://www.w3.org/TR/owl-features/) was developed to interface with ontological databases in a way such that concepts are not simply stored but also assigned meaning in the DB.

In the end, however, no one implementation emerged a victor over the others, and the world is still at large for a universal ontology.

Most modern applications will employ domain-specific ontologies, like the following one for neurological diseases:



(https://code.google.com/p/neurological-disease-ontology/)

❓ So where do we get ontologies to begin with?

Ontologies are nice because they allow us to reason away from specifics and talk instead about **categories**, which doesn't limit our search for an item of interest until necessary.

If we want to make inferences about knowledge at the category level, then we typically want to enforce some notion of inheritance and classes.

ℹ A **class** is simply a categorization that establishes a set relationship of inheritance; all subclasses of a class "inherit" their ancestors' categorizations.

ℹ A **taxonomy** is an ontology where classes and subclasses organize categories hierarchically.

```
; We can represent classes in FOL under
; the assumption of set mechanics
Mammals ⊆ Animals → Subset(Mammals, Animals)
Dogs ⊆ Mammals → Subset(Dogs, Mammals)

; Since we have a taxonomy class structure like the
; above, we can infer the following predicates:
Subset(Dogs, Mammals) ; true because Dogs ⊆ Mammals
Subset(Dogs, Animals) ; true because Dogs ⊆ Mammals ⊆ Animals
```

Nothing surprising there!

This is simple class inheritance like I'm sure you did in some intro programming class... just... for knowledge now!

The real power is when we start exploiting FOL operations, such as implication, which can allow us to extract new facts from our ontologies.

```
; We can use quantification to talk about members of
; our categories
∀x (x ∈ Dogs) ⇒ CanBark(x)

; Saying the above without using set notation, but instead
; FOL:
Member(x, Dogs) ⇒ CanBark(x)
```

We also define some rules to talk about relationships between classes in our ontologies.

⚙ Two or more categories are **disjoint** if they have no members in common.

```
; Examples of disjoint property:
Disjoint({Animals, Rocks}) ; sorry, no pet rocks
Disjoint({Cats, Dogs, Humans}) ; counterexample: Catdog?
```

Some class sets have a special property that they cover the entire field of membership possibilities.

I.e., if you're a member of class A, and there is an exhaustive decomposition of class A into 3 subclasses B, C, and D, then belonging to A means you *must* belong to one of B, C, or D (or multiple)

⚙ An **exhaustive decomposition** defines a set of subclasses that, together, exhaustively compose some other class.

⚙ A **partition** is an exhaustive decomposition where the subclasses are also disjoint.

```
; Exhaustive decompositions:
ExDec({Americans, Canadians, Mexicans}, NorthAmericans)

; i.e., if you are a NorthAmerican, then you must be at least
; an American, Canadian, or Mexican (or multiples!)


; Partitions:
Partition({Even, Odd}, Numbers)

; i.e., if you are a number, then you are either even or odd,
; but cannot be both even and odd!
```

There are a number of other cool FOL tricks we can do with ontologies to represent different human-parsable concepts (even the abstract ones), including the following:

```
; Physical Parts:
PartOf(California, USA)
PartOf(Hawaii, USA)

; Properties of membership
PartOf(x, USA) ⇒ MURICAN(x)

; Generic aggregations to denote collections
BunchOf({Apple1, Apple2, Apple3})
∀x x ∈ s ⇒ PartOf(x, BunchOf(s))

; Measurements
Temperature(LA, Today) = Farenheit(100)
; Measurement Identities
d ∈ Days ⇒ Duration(d) = Hours(24)

; Mental Events / Knowledge Transfer
Knows(Descartes, Exists(Descartes)) ; philosophy jokes?
Knows(Andrew, ¬CanFly(Pigs))
; Useful for programming our agents with what they
; think that others think 0_o

; Semantic implications
Member(John, OneLeggedPersons)
∀ x x ∈ Persons ∧ (x ≠ John) ⇒ Legs(x, 2)
; Useful for NLP systems to unpack sentences
```

And that's pretty much Knowledge Representation in a nutshell!

The book has some more examples and details but the gist is captured above... nothing really surprising!

Quantifying Uncertainty

Yes, I know, I'm excited too! Finally some probability stuff!

Someone made a very astute observation when we were discussing our propositional logic example with rain making the sidewalk wet:

They asked, "We have these general rules that the sidewalk will always be wet if it's raining, but what if the sidewalk is covered by a tree during the storm?"

Clearly there are exceptions to our general rules, and it's often difficult to represent the infinite number of exceptions that could happen with FOL:

```
  ; The naive reasoner about a wet sidewalk:
  KB =
     1. Weather(Rain)
     2. Weather(Rain) ⇒ Wet(Sidewalk)

  ; Handling exceptions:
  KB =
     1. Weather(Rain)
     2. (Weather(Rain) ∧ ¬Covered(Sidewalk) ∧ ¬WaterResistant(Sidewalk) ∧ ...) ⇒ Wet(Sidewal
k)
```

This brittle KB and reasoning system has several main issues:

⚠ The list of rules for getting the sidewalk wet is **incomplete**; e.g., what if sprinklers turned on and wet the sidewalk but it wasn't raining?

⚠ The rule for getting the sidewalk wet (in this case, only in the event that it's raining) does not handle **exceptions**; e.g., a covering of the sidewalk when it's raining.

⚠ We deal only with boolean logic and have no parsimonious means of representing **uncertainty**; e.g., our sprinklers will turn on only if a coin flip comes up heads.

❓ So, for example, viewing our KB as stated above, does knowing anything about whether or not the sidewalk is wet tell us whether or not it's raining?

ℹ In comes **probabilistic reasoning**, which attempts to model **uncertainty** about our environment in a parsimonious, statistical representation.

ℹ **Uncertainty** about an environment arises when our agent has partial sensor information, any sort of ignorance, or the problem is nondeterministic, but we still must make a rational decision.

So, we're going to attempt to frame our set of possible worlds, just like in propositional logic, in terms of probabilities!

Probabilistic Logic

Probabilistic logic is still interested in the set of possible worlds, but unlike propositional logic, our variables of interest may have more than 2 values (usually called **events**).

For example, instead of just `Raining` and `¬Raining`, we might have a Weather variable with possible values of {Sunny, Raining, Fog}.

Let's use our rain and sidewalk wetness example to spearhead the discussion on probabilistic logic... as a reminder:

```
; We had two variables defined as:
R = Whether or not it is raining
S = Whether or not the sidewalk is wet
```

The fact that I have two propositional variables of interest means that I have 4 possible worlds consisting of:

World	R	S	Interpretation of (R ∧ S)
W0	F	F	It isn't raining and the sidewalk isn't wet

World	R	S	Interpretation of (R ∧ S)
W1	F	T	It isn't raining but the sidewalk IS wet
W2	T	F	It IS raining, but the sidewalk isn't wet
W3	T	T	It IS raining, and the sidewalk IS wet

As we already said, enumerating all possible worlds (with their exceptions and managing uncertainty) is intractable, so instead we'll replace our KB with probability tables.

A **probability table** on some number of variables {A, B, C, ...} defines the probabilities of seeing each possible world, i.e., the chance of that world / combination of variables being observed.

So let's look at a probability table for our (now expanded) example:

Assumptions about our problem:

- We'll define variable W to be the weather which can attain one of 3 values: {sunny, foggy, rainy}
- S will remain a boolean variable of whether or not the sidewalk is wet, but we'll use possible values {wet, dry}.
- We assume we live in a fairly wet environment where fog and rain aren't uncommon (relevant later)

W	S	Pr(W, S)	Explanation
sunny	dry	Pr(sunny, dry) = 0.20	It's sunny and the sidewalk is dry.
sunny	wet	Pr(sunny, wet) = 0.05	It's sunny and the sidewalk is wet (sprinkler maybe?)
foggy	dry	Pr(foggy, dry) = 0.10	It's foggy and the sidewalk is dry (not a heavy fog?)
foggy	wet	Pr(foggy, wet) = 0.10	It's foggy and the sidewalk is wet (collection of dew?)
rainy	dry	Pr(rainy, dry) = 0.15	It's rainy and the sidewalk is dry (covered by tarp?)
rainy	wet	Pr(rainy, wet) = 0.40	It's rainy and the sidewalk is wet (duh)

ℹ We call this table the **joint probability table** for our variables W and S because it represents all possible worlds in our state space with the probabilities of seeing those variable instantiations together.

ℹ Note: Joint probability tables describe the probabilities of seeing the variable instantiations of each world **without any other evidence.**

This is important because it means that a world's chance of being observed can be a consequence of any of that world's variables.

For example, a 5% chance that we see rainy weather and a wet sidewalk could mean that it rarely rains, or when it does rain, the sidewalk rarely gets wet (or any combination of explanations).

Notation

One of the hardest parts of dealing with probabilistic logic is its notation. We'll cover some basics now.

Element	Example	Description
Variable	A, B, C Weather, Sidewalk	**Capitalized** letters, words. These will acquire values in each world of the model
Value	a, b, c sunny, wet Weather = sunny, Sidewalk = wet	**Lowercase** letters, words. These are the instantiations that a given variable may attain.
Table / Distribution	Pr(A, B, C)	**Pr(...)** with at least one uninstantiated variable. These define the probability mass amongst the variables listed in the Pr(...) statement.
World	Pr(a, b, c) Pr(A = a, B = b, C = c)	**Pr(...)** with all values / instantiated variables. These are a single table-row; they define one instantiation of variables.

ℹ There are several **axioms of probability theory** that place intuitive constraints on our distributions / tables:

- The probability that we assign to a given world w must range between 0 (certain not to occur) and 1 (certain to occur), i.e.:

$$0 \leq Pr(w) \leq 1$$

- The sum of all probability values in the **joint distribution** must sum to 1, i.e.:

$$\sum_{w} Pr(w) = 1$$

- The probability of some logical sentence α is just the sum of the probabilites of worlds that satisfy α

$$Pr(\alpha) = \sum_{w \in M(\alpha)} Pr(w)$$

Example
☑ Knowing that last rule, what is the probability that the sidewalk is wet? i.e., α = {S = wet}

We can look at our table and find all of the rows consistent with α, and then simply sum up their individual probabilities:

W	S	Pr(W, S)
sunny	dry	Pr(sunny, dry) = 0.20
sunny	wet	Pr(sunny, wet) = 0.05
foggy	dry	Pr(foggy, dry) = 0.10
foggy	wet	Pr(foggy, wet) = 0.10
rainy	dry	Pr(rainy, dry) = 0.15
rainy	wet	Pr(rainy, wet) = 0.40

There are also some useful properties of our notion of worlds that transfer to probability theory:

For example, we can arbitrarily slice up the set of possible worlds such that:

Ω is the set of all possible worlds.

$$M(\alpha) \cup M(\neg\alpha) = \Omega$$

Therefore, if:

$$Pr(\alpha) = \sum_{w \in M(\alpha)} Pr(w)$$

Then

$$Pr(\neg\alpha) = \sum_{w \in M(\neg\alpha)} Pr(w)$$

And since we know the sum of all rows must add to 1, then:

$$Pr(\alpha) = 1 - Pr(\neg\alpha)$$

❓ If Pr(rainy) = 0.55, then what's the Pr(¬rainy)?

Simple, right?

Here's another useful property:

⚙ The **inclusion-exclusion principle** says that the probability of sentence α disjoined with sentence β is the sum of Pr(α) and Pr(β) minus their overlap (i.e., Pr(α ∧ β)).

$$Pr(\alpha \vee \beta) = Pr(\alpha) + Pr(\beta) - Pr(\alpha \wedge \beta)$$

Example
☑ In our example (repeated below), what is Pr(W = rainy ∨ S = wet)?

W	S	Pr(W, S)
sunny	dry	Pr(sunny, dry) = 0.20
sunny	wet	Pr(sunny, wet) = 0.05
foggy	dry	Pr(foggy, dry) = 0.10

W	S	Pr(W, S)
foggy	wet	Pr(foggy, wet) = 0.10
rainy	dry	Pr(rainy, dry) = 0.15
rainy	wet	Pr(rainy, wet) = 0.40

Two sentences α and β are said to be **mutually exclusive** if:

$$Pr(\alpha \wedge \beta) = 0$$

We can bring this back to our inclusion-exclusion principle and observe that the "overlap" of probability mass for two logical sentences is 0 for any mutually exclusive sentences.

In our example above, what is Pr(W = rainy $\wedge$ W = foggy)?

In our example above, what is Pr(W = rainy $\vee$ W = foggy)?

The **law of total probability** says that for some set β_i of pair-wise, mutually exclusive sentences that partition Ω (cover every possible world), then:

$$Pr(\alpha) = \sum_i Pr(\alpha, \beta_i)$$

The law of total probability says that "If I want to determine the probability of α, sum over every world where α and some sentence β_i (from my mutually exclusive set of sentences β that partition the whole state space) are found together."

We can use the law of total probability to reduce larger, more general joint distributions into smaller, more specific distributions or worlds of more particular interest.

Example

Let's try that using our running example to find the probability that the sidewalk is dry...

Defining β

$$\beta = \{W = sunny, W = foggy, W = rainy\}$$

Does our β partition Ω?

Yes! The weather must be sunny, foggy, or rainy in our state space

Is each β_i pairwise mutually exclusive of the other?

Yes! The weather can't be both sunny and foggy, etc.

Therefore, to determine the probability that the sidewalk is dry:

$$\alpha = S = dry$$

We use the law of total probability to derive:

$$\begin{aligned} Pr(\alpha) &= Pr(sunny, dry) + Pr(foggy, dry) + Pr(rainy, dry) \\ &= 0.20 + 0.10 + 0.15 \\ &= 0.45 \end{aligned}$$

A reasonable question you might be asking is: where did these probability numbers come from in the first place?

This remains a hotly debated topic and the answer is: it depends who you ask!

Here are some of the popular opinions:

- **Frequentists** claim that you must run experiments to discover the probability values, e.g., in our problem, counting the number of days in a 365 day calendar year how many had both rain and a wet sidewalk, etc.
- **Objectivists** believe that the probabilities are real aspects of the way the universe operates, and the experiments of the frequentists attempt to measure these universal mechanisms.
- **Subjectivists** believe that probabilities represent an agent's beliefs about a system, and so the values are simply their prior educated guesses (that should be updated via Bayesian approaches, which we'll discuss later, if any evidence is brought before them)

Priors are the probabilities we assign to certain worlds before we take any circumstantial evidence into account.

The above schools of thought are, therefore, simply different ways to get our priors!

The values in our table are priors as well because they are our understanding of the chances of witnessing the listed combination of variable instantiations without knowing anything more.

...but what if we do know something more 0_o

Updating Beliefs and Conditioning

Say we started with our joint probability table (the priors) but suddenly, hark! We witness that it's raining outside!

As soon as we **observe evidence** about some variable in our environment, we want to take that observation into account.

Conditioning is an operation from probability theory that allows us to update our beliefs given evidence about the current state of our environment.

Let's say we witnessed that it was raining outside...

Conditioning says, "I don't care about all of those other probabilities for when it's sunny or foggy out... it's raining! I'm looking at it rain right now!"

We say that evidence witnessed is now **given** in our probability calculations.

For some query sentence α and witnessed evidence β, we denote the probability of witnessing α under evidence β as:

$$Pr(\alpha|\beta)$$

In terms of our joint probability distribution, conditioning on some evidence β means the chance of witnessing a world that is **inconsistent** with our evidence β is 0. Formally, for witnessed evidence β:

$$Pr(\neg\beta) = Pr(\alpha, \neg\beta) = Pr(\alpha|\neg\beta) = 0$$

We can interpret the above as saying, "No world that is inconsistent with what we've witnessed shall be considered possible."

And so, we assign such contrary worlds a probability of 0% given the evidence.

ℹ Note: we may represent joint distributions using commas instead of the logical-and operator, e.g.:

$$Pr(\alpha \wedge \beta) = Pr(\alpha, \beta)$$

But remember, if we're zeroing out some rows that are inconsistent with our evidence, then what's left still has to add up to 1!

⚙ Conditioning requires **normalization** of worlds consistent with the evidence such that (for some normalizing constant N):

$$Pr(\alpha|\beta) = N * Pr(\alpha \wedge \beta)$$

Where our normalizing constant is simply 1 over the sum of probability mass across worlds consistent with the evidence:

$$N = \frac{1}{\sum_{\beta_i \in M(\beta)} Pr(\beta_i)}$$

Example
☑ Compute the probability that the sidewalk is wet **given** that it is raining (Pr(S = wet | W = rainy)) by first finding the probability of each world under the evidence W = rainy:

W	S	Pr(W, S)	Pr(S | W = rainy)
sunny	dry	0.20	???
sunny	wet	0.05	???
foggy	dry	0.10	???
foggy	wet	0.10	???

W	S	Pr(W, S)	Pr(S | W = rainy)
rainy	dry	0.15	???
rainy	wet	0.40	???

(1) Zero out the rows inconsistent with our evidence (click for solution).

(2) Normalize the remaining rows consistent with our evidence.

So there we have it! There's about a 73% chance that the sidewalk is wet if we observe that it's raining out.

Handily, there's also a closed form for conditioning (among others) that is doing what we just did above:

Bayes' Conditioning is a closed-form for computing a conditional quantity in terms of a joint, given by:

$$Pr(\alpha|\beta) = \frac{Pr(\alpha \wedge \beta)}{Pr(\beta)}$$

Hey! Look at that! Turns out our normalizing constant was actually just $\frac{1}{Pr(\beta)}$!

Makes sense, right? Because:

$$Pr(\beta) = \sum_{\omega \in M(\beta)} Pr(\omega)$$

Conditioning is great because it allows us to update our probability distributions as soon as we reduce our uncertainty through observation in some capacity.

Independence

⚙ Events α and β are said to be **independent** if knowing something about one tells us nothing about the other. Formally, for query α and evidence β, we say that α is independent from β and write $\alpha \perp\!\!\!\perp \beta$ whenever:

$$Pr(\alpha|\beta) = Pr(\alpha)$$

This makes sense because if I was curious about the probability that the sidewalk was wet α = {S = wet}, and you told me that you just saw the incredible hulk driving a truck outside β = {H = drivingTruck}, then I might be intrigued but:

$$Pr(S = wet|H = drivingTruck) = Pr(S = wet)$$

Intuitively, we see that learning about the incredible hulk's driving does not change the probability of the sidewalk being wet.

In this sense, we can call independence a measure of **relevance** of some variables to one another.

We'll do some examples with this later...

Chain Rule

As we saw above, Bayes' conditioning tells us that:

$$Pr(\alpha|\beta) = \frac{Pr(\alpha, \beta)}{Pr(\beta)}$$

There are a number of other tricks we can play with it to derive some interesting results!

Starting with Bayes' conditioning, what happens when we multiply our normalizing constant $Pr(\beta)$ to both sides?

$$Pr(\alpha|\beta) = \frac{Pr(\alpha, \beta)}{Pr(\beta)}$$
$$\therefore Pr(\alpha, \beta) = Pr(\alpha|\beta) * Pr(\beta)$$

Re-writing that to pretty it up:

$$Pr(\alpha, \beta) = Pr(\alpha|\beta) * Pr(\beta)$$

We could take this a step further; think about having three variables in the mix! This lets us choose different sets of variables for our α and β:

$$\begin{aligned} Pr(A, B, C) &= Pr(A, B|C) * Pr(C) \\ &= Pr(A, C|B) * Pr(B) \\ &= Pr(B, C|A) * Pr(A) \\ &= Pr(A|B, C) * Pr(B, C) \\ &= Pr(B|A, C) * Pr(A, C) \\ &= Pr(C|A, B) * Pr(A, B) \end{aligned}$$

This re-arranging of Bayes' Conditioning gives us the **chain rule** for factorization such that:

Example 1:

$$Pr(A, B) = Pr(A|B) * Pr(B)$$

Example 2:

$$\begin{aligned} Pr(A, B, C) &= Pr(A|B, C) * Pr(B, C) \\ &= Pr(A|B, C) * Pr(B|C) * Pr(C) \end{aligned}$$

Example 3:

$$\begin{aligned} Pr(A, B, C, D) &= Pr(A|B, C, D) * Pr(B, C, D) \\ &= Pr(A|B, C, D) * Pr(B|C, D) * Pr(C, D) \\ &= Pr(A|B, C, D) * Pr(B|C, D) * Pr(C|D) * Pr(D) \end{aligned}$$

Etc., etc., etc...

Notice one important point from the above factorizations:

What happens if A is independent from B? For example, the chain rule factorization:

$$Pr(A, B) = Pr(A|B) * Pr(B)$$

If A is independent of B, then we know:

$$Pr(A|B) = P(A)$$

Therefore, subbing into our first equation, if A is independent of B:

$$Pr(A, B) = Pr(A) * Pr(B)$$

These points are, of course, all leading us up to the big-daddy of probabilistic reasoning: Bayes' Theorem

Bayes' Theorem

Let's take our Bayes' Conditioning one step further to unlock the true power of Bayesian reasoning:

We saw this from Bayes' Conditioning

$$Pr(A, B) = Pr(A|B) * Pr(B)$$

But would you also agree that:

$$Pr(A, B) = Pr(B, A)$$

It is just a conjunction! Sure! That works... So, if:

$$Pr(A, B) = Pr(A|B) * Pr(B)$$

...and:

$$Pr(B, A) = Pr(B|A) * Pr(A)$$

...AND:

$$Pr(A, B) = Pr(B, A)$$

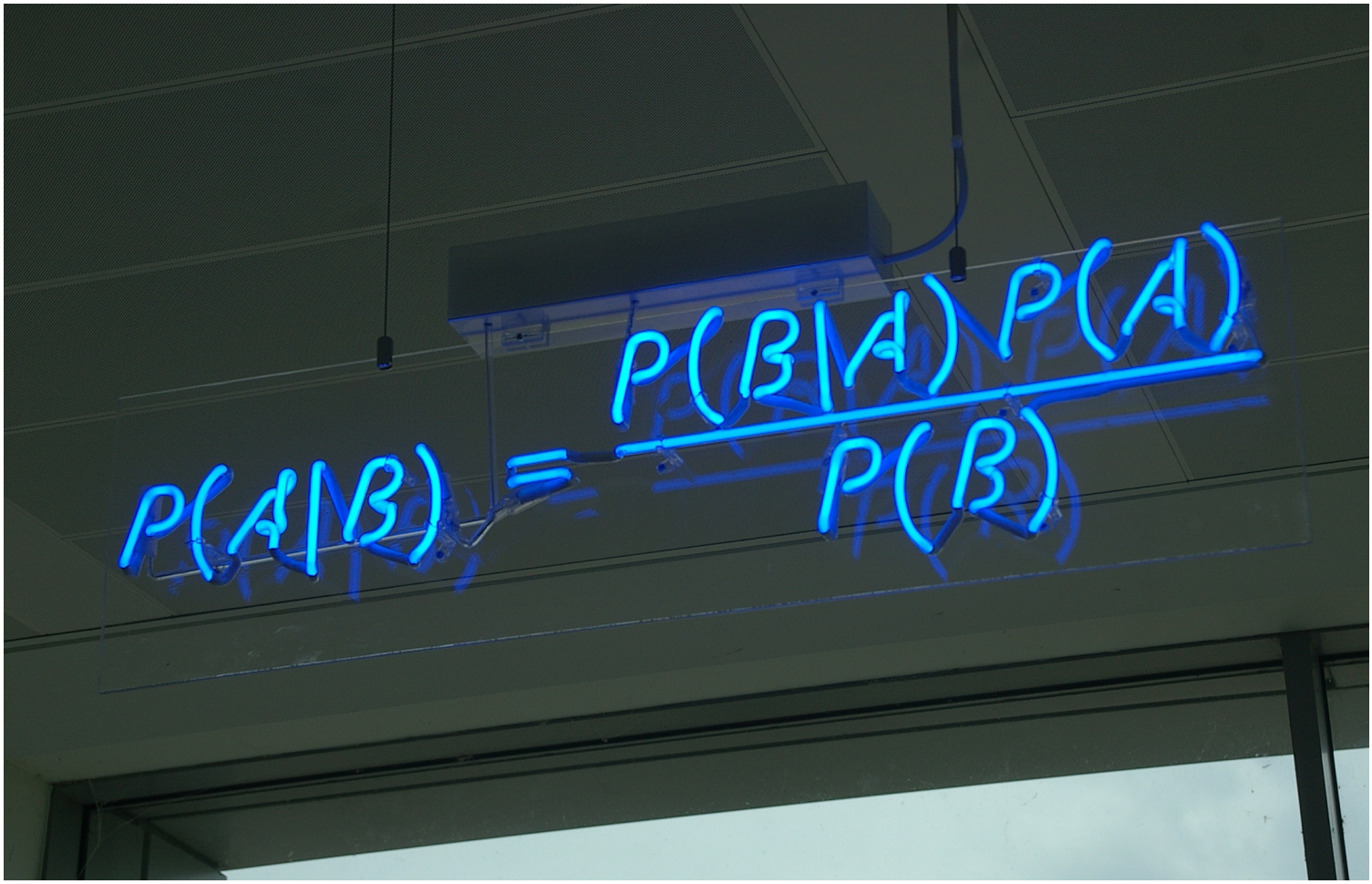
...THEN...

$$Pr(A|B) * Pr(B) = Pr(B|A) * Pr(A)$$

...AND SO!!!

$$Pr(A|B) = \frac{Pr(B|A) * Pr(A)}{Pr(B)}$$

This is Bayes' Rule... and yeah... it's kinda a big deal

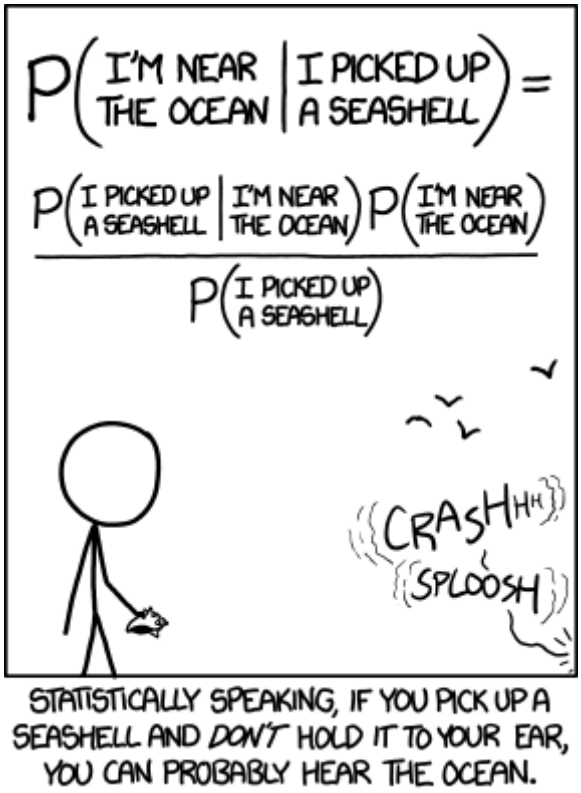


...at least some people seem to think so...

❓ What is so damn exciting about Bayes' theorem? Hint: consider P(E | C) where E is an effect and C is a cause of that effect.

This is huge because we may not know anything about P(A | B), but knowing something about P(B | A) gives us a wealth of information that we may now infer.

...and of course there's a relevant XKCD, thanks for asking!



(http://xkcd.com/1236/)

Example

Explain the above XKCD!

Example

Use Bayes' Theorem (replicated below) to compute the following quantity:

$$Pr(A|B) = \frac{Pr(B|A) * Pr(A)}{Pr(B)}$$

Let's pretend we had a different joint distribution than our one listed above such that we *only* knew the following:

$$Pr(S = wet) = 0.60$$
$$Pr(W = rainy) = 0.40$$
$$Pr(S = wet|W = rainy) = 0.75$$

What if my new question is: what is the probability that it's raining, given that the sidewalk is wet?

$$Pr(W = rainy|S = wet)$$

Click for solution.

Applying Bayes' Theorem

We've learned the probabilistic theory and mechanisms behind Bayes' Theorem, so let's see how it all fits together.

Shall we start with a motivating example? Don't feel bad if you don't get this at first, about 85% of medical doctors asked the same question get it wrong.

Example
Read the following problem description, and for each probability mentioned, formalize it into a Pr(x | y) statement and then solve for the correct quantity. Assume two binary variables: Test and Disease, as described by the problem.

> A very rare condition, Schistosoforneymiosis, is found in about 1/1000 of those tested for it; sufferers experience a consistently wet left foot and sentient freckles.
>
> The test is an elaborate procedure involving multiple probes, and returns an end result that is either positive (test) or negative (¬test). The problem is that the tests are not perfect, but 95% of people who have the disease will test positive, and 2% of people who do *not* have the disease will test positive.
>
> **If a patient tests positive, what is the probability that they have the disease?**

Let's dissect this problem and determine the proper quantities for each component to lead to our solution.

Translate the sentence into a Pr statement: "A very rare condition, Schistosoforneymiosis, is found in about 1/1000 of those tested for it."

The above represents the **prior** probability that we discussed from last time, i.e., the chance that someone has the condition before accounting for any evidence.

Translate the sentence into a Pr statement: "95% of people who have the disease will test positive."

This value represents the case of a **true positive** since the test is positive and so is the disease.

❓ Translate the sentence into a Pr statement: "2% of people who do *not* have the disease will test positive."

This value represents the case of a **false positive** since the test is positive but the disease is not.

❓ Translate the sentence into a Pr statement: "If a patient tests positive, what is the probability that they have the disease?"

Alright, off to a good start! Let's write out everything that we have so far:

$$\begin{aligned} Pr(disease) &= 0.001 \\ Pr(test|disease) &= 0.95 \\ Pr(test|\neg disease) &= 0.02 \\ Pr(disease|test) &= ??? \end{aligned}$$

So, if it wasn't obvious before, we need to use Bayes' Theorem in order to solve for our target quantity! Let's write out what we'll need for that:

$$Pr(disease|test) = \frac{Pr(test|disease) * Pr(disease)}{Pr(test)}$$

Do we have everything that we need to solve for Pr(disease | test)?

Well, yes and no; we have everything we need to calculate what we need, explicitly. In particular, we have everything on the RHS but Pr(test).

Do we have a means of calculating the Pr(test)?

❓ Click here if you need a hint on how to calculate Pr(test)...

"But Andrew, we don't have things in terms of Pr(α, β), only **conditions**!"

You're correct! You even said the strategy we need to use! Let's take a look:

I claim that the choice for β is a partition in which each β_i is mutually exclusive, do you agree?

$$\beta = \{disease, \neg disease\}$$

So, by the Law of Total Probability:

$$\begin{aligned} Pr(\alpha) &= \sum_i Pr(\alpha, \beta_i) \\ Pr(test) &= \sum_i Pr(test, \beta_i) \\ &= Pr(test, disease) + Pr(test, \neg disease) \end{aligned}$$

Now, we can condition in order to get:

$$\begin{aligned} Pr(Test) = Pr(test|disease) * Pr(disease) \\ + \ Pr(test|\neg disease) * Pr(\neg disease) \end{aligned}$$

We do not *explicitly* have one of these quantities above, BUT, obvserve:

$$\begin{aligned} Pr(\neg disease) &= 1 - Pr(disease) \\ &= 1 - 0.001 \\ &= 0.999 \end{aligned}$$

Therefore:

$$Pr(test) = 0.95 * 0.001 + 0.02 * 0.999 \approx 0.02093$$

And now, plugging back into our original:

$$\begin{aligned} Pr(disease|test) &= \frac{Pr(test|disease) * Pr(disease)}{Pr(test)} \\ &\approx \frac{0.95 * 0.001}{0.02093} \\ &\approx 0.045 \end{aligned}$$

Wow! That's a really small chance given that our incredibly accurate test was still positive!

Understanding Bayes' Theorem gives us a lot of power to detect non-obvious consequences like the above.

Next week, we'll look at a specific application to Bayesian networks... a beautiful data structure for reasoning under uncertainty.

Homework 3

Previous to HW3, we weren't really sure how to root the notion of frames to first order logic... now we can take the time to gain some clarity:

FoL Concept	FoL Example	Frame Equivalent	Frame Example
Constant	Andrew	0-slot frame	(ANDREW)
Predicate	Teaches(Andrew, CS161)	Frame	(TEACHES AGENT (ANDREW) OBJECT (CS161))
Variable	Teaches(Andrew, x)	Variable	(TEACHES AGENT (ANDREW) OBJECT (V X))
Function	Awesome(TeacherOf(161))	Gap	(AWESOME TEACHER-OF CS161)

Here are a couple of hints:

Function	Hint
UNIFY-FR	The best way to conquer this problem is to delegate the labor for each of the 3 input types: • Variables: have a function that performs the variable binding and also checks for binding conflicts (the book's UNIVAR handles this, if you can copy its approach!) • Frames: have a function that searches for variables requiring binding in your frames, which then delegates the work to your variable unifying function. Remember to also verify that predicates match, though you may well do this in the parent UNIFY-FR function. • Lists of Frames: have a function that searches for successful unifications between frames in the first list with some subset of frames in the second. Beware of red-herring unifications which might make a binding that makes future unifications unsuccessful! You can combat that with some clever recursion, hint: think about removing red-herring frames in the second list (you might do this in UNIFY-FR)!
SUBST-FR	Fairly trivial; resembles two functions we've already done with only minor modification.

Function	Hint
MP-INFER	Remember that you must successfully unify across ALL of a rule's premises in order to infer its conclusion. Furthermore, be wary that variable names in rules have already been standardized, meaning a given variable cannot have two separate bindings. Fairly trivial once UNIFY-FR is complete.
FRW-CHAIN	Straightforward once you've completed RR-FRW. The "hardest" part is determining when you've inferred a new conclusion or not. I suggest making a helper function for this purpose.