

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!

á **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?!

Example

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

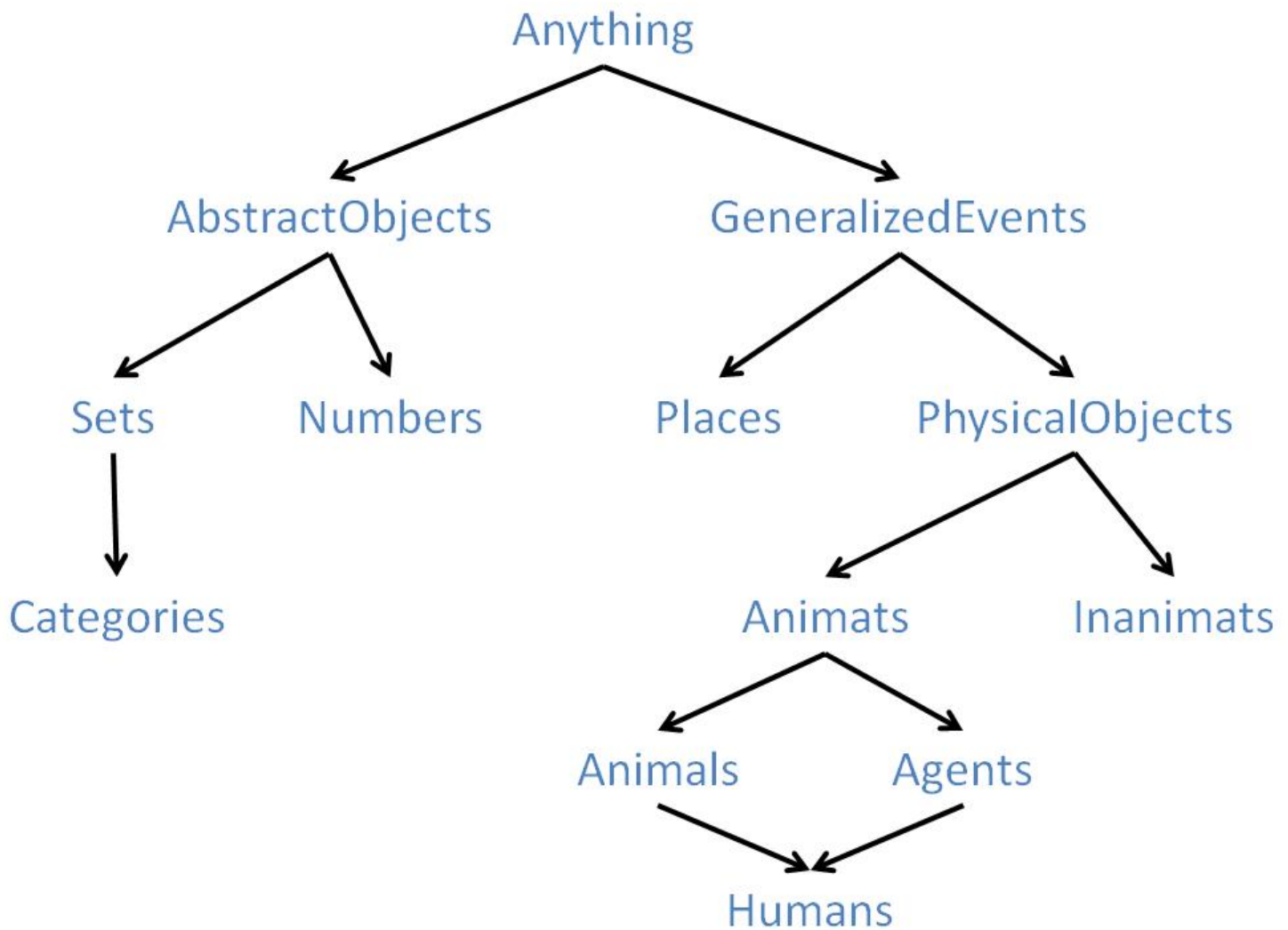
So, naturally, we want the information that our computer spits back to be... well... natural, and readable to a human!

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

á **An ontology** is a classification system (usually organized into Knowledge Bases) such that we establish membership of specific world entities into

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.



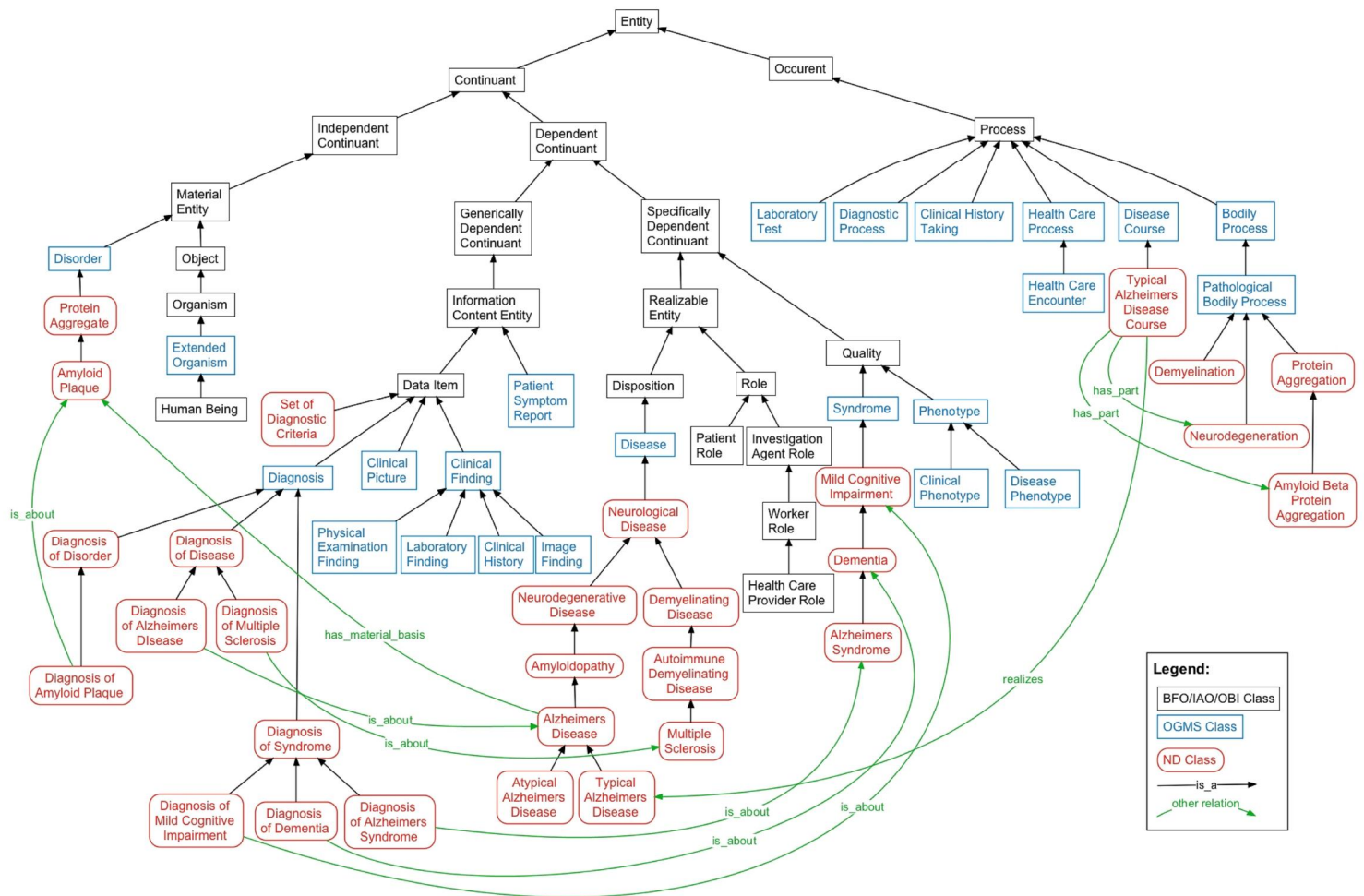
In practice, as you might've expected, creating and then deciding upon a unified ontology has been empirically difficult.

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/>)

à 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.

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

đ 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  $\subseteq$  Animals  $\rightarrow$  Subset(Mammals, Animals)
Dogs  $\subseteq$  Mammals  $\rightarrow$  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  $\subseteq$  Mammals
Subset(Dogs, Animals) ; true because Dogs  $\subseteq$  Mammals  $\subseteq$  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
 $\forall x (x \in \text{Dogs}) \Rightarrow \text{CanBark}(x)$ 

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

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

H 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)

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

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

```
; Exhaustive decompositions:  
ExDec({Americans, Canadians, Mexicans}, NorthAmericans)  
  
; i.e., if you're a NorthAmerican, then you must be at least  
; an American, Canadian, or Mexican (or multiples!)  
  
; Partitions:  
Partition({Even, Odd}, Numbers)  
  
; i.e., if you're a number, then you're 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)  $\Rightarrow$  Wet(Sidewalk)

; Handling exceptions:
KB =
  1. Weather(Rain)
  2. (Weather(Rain)  $\wedge$   $\neg$ Covered(Sidewalk)  $\wedge$   $\neg$ WaterResistant(Sidewalk)  $\wedge$  ...)  $\Rightarrow$  Wet(
Sidewalk)

; Very difficult and impractical to cover all the possible
; Cases where it's raining and the sidewalk isn't necessarily wet!

; Also, our intelligent agent just might not *know* whether the
; sidewalk is covered
```

Another issue is the following: 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?

No! We don't have any rules to infer about observing the sidewalk whether it's raining or not, but certainly a wet sidewalk *suggests* that it might be raining.

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

á **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 \neg **Raining** , we might have a Weather variable with possible values of {Sunny, Raining, Fog}.

H In probabilistic logic, variables set to a given value are represented as **Var** = val where Var is some variable and val is its current value.

NOTE: The convention is to Capitalize variable names while keeping the variable values in lowercase.

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's 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 ($R \wedge S$)
W0	F	F	It isn't raining and the sidewalk isn't wet
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

Our query asked if the sidewalk was wet, i.e., $\alpha = S$, dictated by the rule:

```
KB =  
  ; If it's raining, then the sidewalk is wet  
  1.  $R \Rightarrow S$   
  
  ; It is raining!  
  2. R
```


This meant that we could ascertain the models of our world (in this case, $M(\alpha) = \{W3\}$) such that:

World	R	S	$(R \Rightarrow S) \wedge R$
W0	F	F	F
W1	F	T	F
W2	T	F	F
W3	T	T	T

This brittle KB and reasoning system has two 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.

As we already said, enumerating all of these possibilities is intractable, so instead we'll replace our KB with probability tables.

á A **probability table** on some number of variables $\{A, B, C, \dots\}$ defines the probabilities of each possible world, i.e., the chance of that world happening.

Let's talk a little about notation and then we'll see how this works.

ℋ **$\text{Pr}(A, B, C)$** (uppercase variables) refers to a **probability table** of probabilities for variables A, B, and C.

ℋ **$\text{Pr}(A = a, B = b, C = c)$** refers to a **row of that probability table** such that variable A attains value a, B attains value b, etc. Setting a variable X to some value x is called a **variable instantiation**.

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

á 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.

There are a few things to notice immediately about our joint probability table:

- 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.:

$$0 \leq \text{Pr}(w) \leq 1$$

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

$$\sum_w \text{Pr}(w) = 1$$

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

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

Example

^t Knowing that last rule, what is the probability that the sidewalk is wet? i.e., $\alpha = \{S = \text{wet}\}$

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:

```

; i.e.,  $\Omega$  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:
 $\Pr(\alpha) = 1 - \Pr(\neg\alpha)$ 

```

Example

^t If $\Pr(\text{rainy}) = 0.55$, then what's the $\Pr(\neg\text{rainy})$?

Simple, right?

Here's another useful property:

^H The **inclusion-exclusion principle** says that the probability of sentence α disjoined with sentence β is the sum of $\Pr(\alpha)$ and $\Pr(\beta)$ minus their overlap (i.e., $\Pr(\alpha \wedge \beta)$).

Numerically, the inclusion-exclusion principle looks like:

```

; Inclusion-Exclusion Principle:
 $\Pr(\alpha \vee \beta) = \Pr(\alpha) + \Pr(\beta) - \Pr(\alpha \wedge \beta)$ 

```

Example

^t In our example above, what is $\Pr(W = \text{rainy} \vee S = \text{wet})$?

^H 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" component is 0 for any mutually exclusive sentences.

Example

^t In our example above, what is $\Pr(W = \text{rainy} \vee W = \text{foggy})$?

^H The law of total probability says that for some set $\{\beta_i\}$ 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."

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

```
; Defining  $\beta$ 
 $\beta = \{W = \text{sunny}, W = \text{foggy}, W = \text{rainy}\}$ 

; Does our  $\beta$  partition  $\Omega$ ?
; Yes! The weather must be sunny, foggy, or rainy in
; our state space

; Is each  $\beta_i$  pairwise mutually exclusive of the
; other?
; Yes! The weather can't be both sunny and foggy, etc.

; Therefore:
 $\alpha = \{S = \text{dry}\}$ 
 $\Pr(\alpha) = \Pr(\text{sunny}, \text{dry}) + \Pr(\text{foggy}, \text{dry}) + \Pr(\text{rainy}, \text{dry})$ 
          = 0.20 + 0.10 + 0.15
          = 0.45
```

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)

á **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.

á **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, which we represent as:

$\text{Pr}(\alpha \mid \beta)$

; **Where** α **is** still some query sentence
; **and** β **is** whatever evidence we just witnessed

If we're thinking in terms of our joint probability distribution, then conditioning on some evidence essentially causes the chance of witnessing a world that is **inconsistent** with our evidence to go to 0.

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!

So, we perform **normalization** such that:

; **For** some normalizing constant N
 $\text{Pr}(\alpha \mid \beta) = N * \text{Pr}(\alpha \wedge \beta)$

Example

^t Compute the probability that the sidewalk is wet **given** that it is raining out: $\text{Pr}(S = \text{wet} \mid W = \text{rainy})$

First, we can think about zeroing out the rows that are inconsistent with our evidence, $\{W = \text{rainy}\}$

W	S	$\text{Pr}(W, S)$	$\text{Pr}(S \mid W = \text{rainy})$
sunny	dry	0.20	0
sunny	wet	0.05	0
foggy	dry	0.10	0
foggy	wet	0.10	0
rainy	dry	0.15	$0.15 / N$
rainy	wet	0.40	$0.40 / N$

We're not done yet! We have to normalize our remaining two rows consistent with the evidence that it's raining!

```
; To find N, we simply add up the joint
; probabilities of all the worlds consistent
; with the evidence,  $\beta$ :
 $N = \sum_{\alpha} P(\alpha \wedge \beta)$ 

 $N = 0.15 + 0.40$ 
 $= 0.55$ 

; Now, we normalize, giving us:
 $\Pr(S = \text{dry} \mid W = \text{rainy}) = 0.15 / N$ 
 $= 0.15 / 0.55$ 
 $= 0.27$ 

 $\Pr(S = \text{wet} \mid W = \text{rainy}) = 0.40 / N$ 
 $= 0.40 / 0.55$ 
 $= 0.73$ 
```

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
 $\Pr(\alpha \mid \beta) = \Pr(\alpha \wedge \beta) / \Pr(\beta)$ 

; Hey! Look at that! Turns out our normalizing
; constant was actually just  $\Pr(\beta)$ !

; Makes sense, right? Because:
 $\Pr(\beta) = \sum_{\alpha} P(\alpha \wedge \beta)$ 
```

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

H Events α and β are said to be **independent** if knowing something about one tells us nothing about the other.

We can represent this in probability calculus as:

; Independence:

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

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

$$\begin{aligned} & \Pr(S = \text{wet} \mid H = \text{drivingTruck}) \\ &= \Pr(S = \text{wet}) \end{aligned}$$

; Because learning that the incredible hulk
; is driving a truck outside **is** irrelevant to
; the probability that the sidewalk **is** wet

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

As we saw above, $\Pr(\alpha \mid \beta) = \Pr(\alpha \wedge \beta) / \Pr(\beta)$.

This was our Bayes' conditioning example, but there are a number of other tricks we can play with it!

; What happens **when** we multiple **our** normalizing constant,
; $\Pr(\beta)$ to both sides?

$$\begin{aligned} & \Pr(\alpha \mid \beta) = \Pr(\alpha \wedge \beta) / \Pr(\beta) \\ &= \Pr(\alpha \mid \beta) * \Pr(\beta) = \Pr(\alpha \wedge \beta) \end{aligned}$$

; Re-writing that to pretty it up:

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

; 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

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

H This interesting 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:
Pr(A, B, C) = Pr(A, B | C) * Pr(C)
             = Pr(A | B, C) * Pr(B | C) * Pr(C)

; Example 3:
Pr(A, B, C, D) = Pr(A | B, C, D) * Pr(B | C, D) * Pr(C | D) * Pr(D)

; ...

```

Notice one important point from the above factorizations:

What happens if A is independent from B?

```

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)

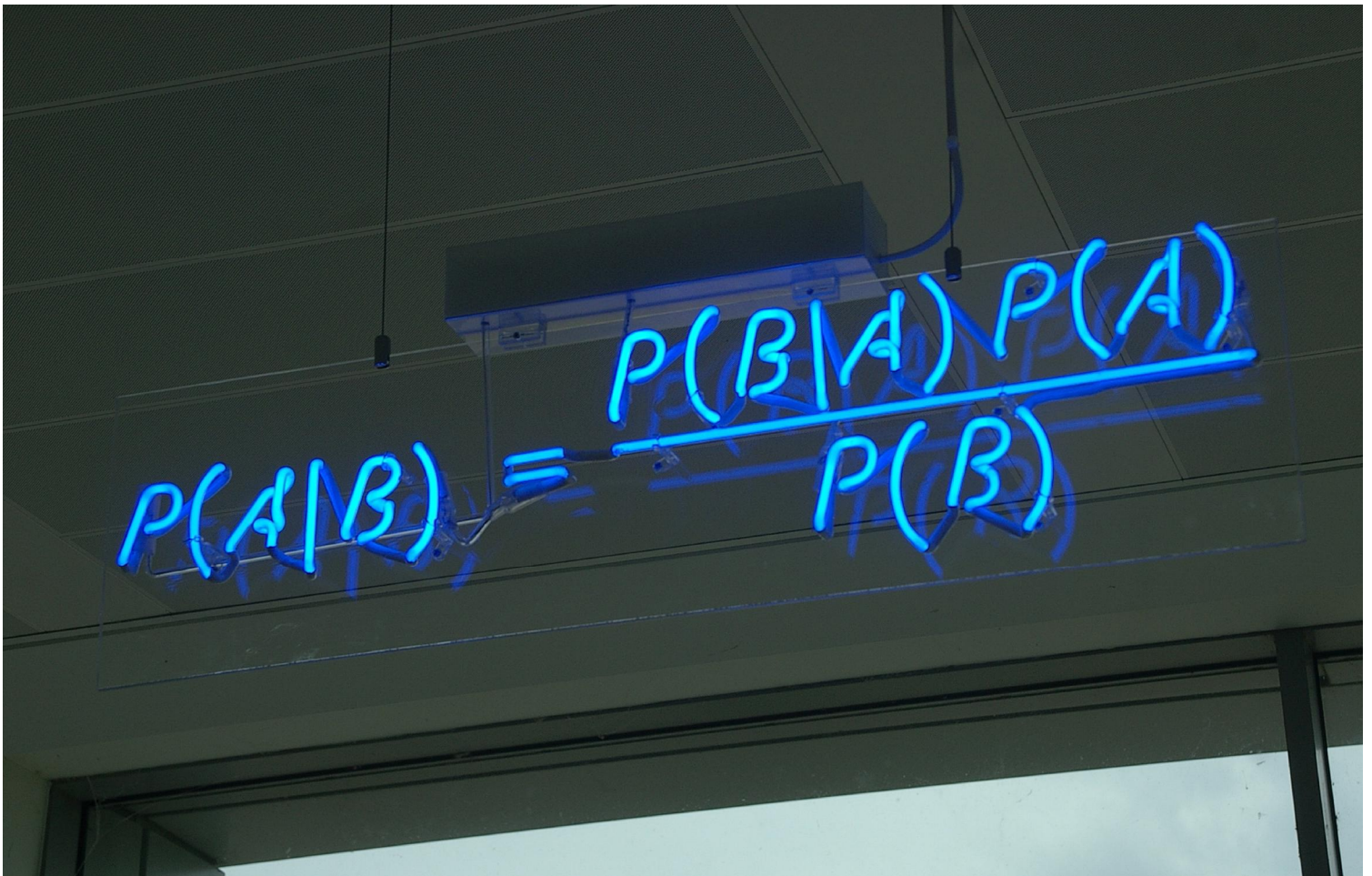
; ...and:
Pr(B, A) = Pr(B | A) * Pr(A)

; ...AND:
Pr(A, B) = Pr(B, A)

; ...THEN...
Pr(A | B) * Pr(B) = Pr(B | A) * Pr(A)

; ...AND SO!!!
Pr(A | B) = (Pr(B | A) * Pr(A)) / Pr(B)

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

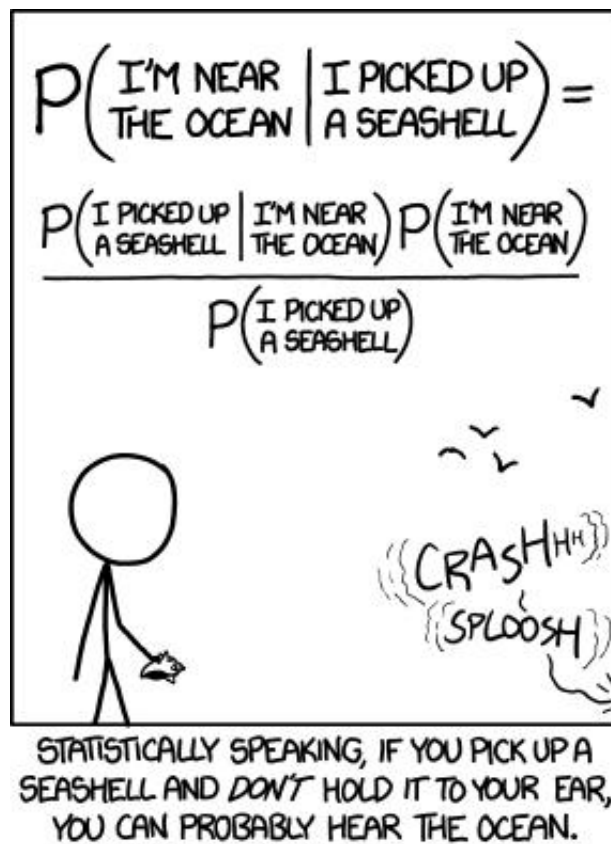


...at least some people seem to think so...

à What is so damn exciting about Bayes' theorem?

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

^t Use Bayes' Theorem to compute the following quantity:

```

; 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) = (Pr(S = Wet | W = rainy) * Pr(W = rainy)) / Pr(S = wet)
                        = (0.75 * 0.40) / 0.60
                        = 0.50

```

Neat! Next class we'll explore the FULL POTENTIAL of Bayes' Theorem...

Homework 4

Here it is everyone... the homework you've all been waiting for!

I hope you really got HWs 1 - 3 because... oh... wait...

No, actually HW4 doesn't use anything from the previous homeworks :P

Previous to HW4, 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)

So, HW4 is going to dive into Variables (which we mentioned in HW1 but only are now seeing in HW4) such that we will perform inference on some KB of frames to accomplish our story parsing.

This assignment uses forward chaining to perform inference, and backward chaining to derive some English sentence answers readable by a human.

We'll take a look at it now!
