

* We would like to have more intelligent assistants such as Data from Star Trek. What is the functionality that's missing in between?
* At an abstract level, one fundamental thing a good personal assistant should be able to do is to take in information from people and be able to answer questions that require drawing inferences from the facts.
* In some sense, telling the system information is like machine learning, but it feels like a very different form of learning than seeing 10M images and their labels or 10M sentences and their translations. The type of information we get here is both more heterogenous, more abstract, and the expectation is that we process it more deeply (we don't want to have to tell our personal assistant 100 times that we prefer morning meetings).
* And how do we interact with our personal assistants? Let's use natural language, the very tool that was built for communication

Natural language

Example:

* A **dime** is better than a **nickel**.
* A **nickel** is better than a **penny**.
* Therefore, a **dime** is better than a **penny**.

Example:

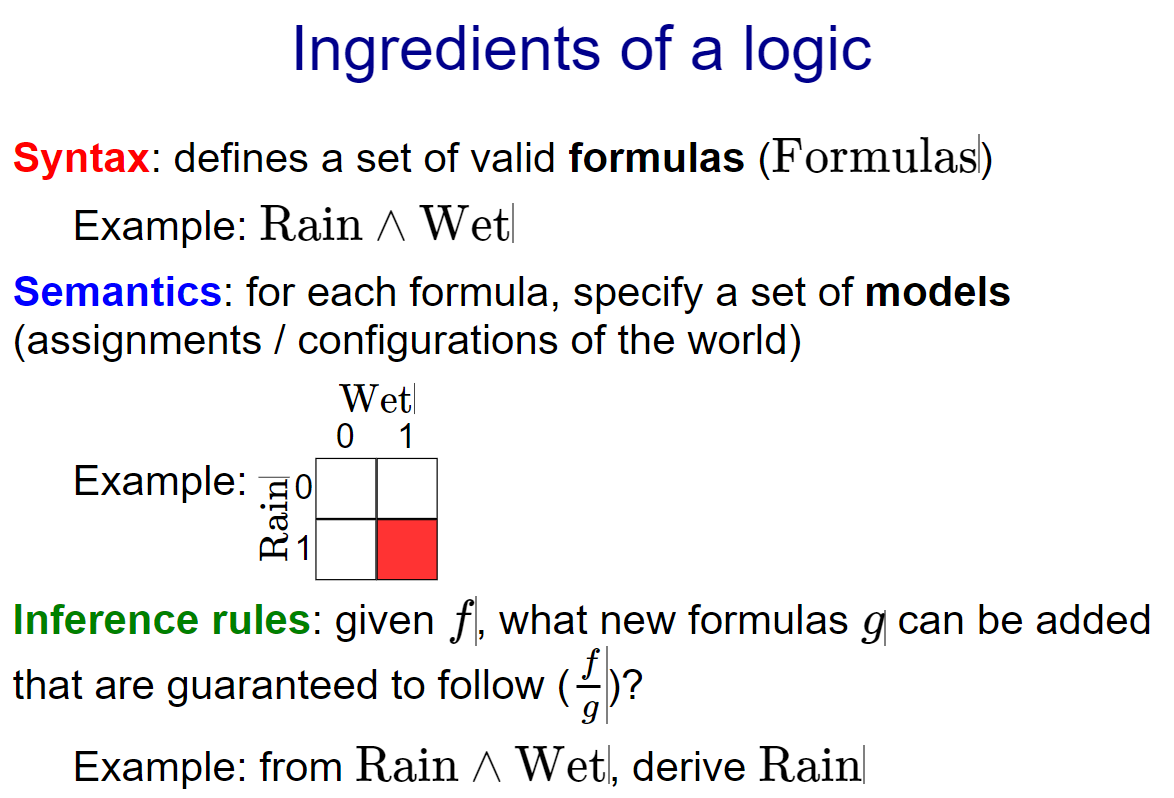
* A **penny** is better than **nothing**.
* **Nothing** is better than **world peace**.
* Therefore, a **penny** is better than **world peace**???

Natural language is slippery...

* But natural language is tricky, because it is replete with ambiguities and vagueness. And drawing inferences using natural languages can be quite slippery. Of course, some concepts are genuinely vague and slippery, and natural language is as good as it gets, but that still leaves open the question of how a computer would handle those cases.



* Let's think about language a bit deeply. What does it really buy you? Primarily, language is this wonderful human creation that allows us to express and communicate complex ideas and thoughts.
* We have mostly been talking about natural languages such as English and German. But as you all know, there are programming languages as well, which allow one to express computation formally so that a computer can understand it.
* This lecture is mostly about logical languages such as propositional logic and first-order logic. These are formal languages, but are a more suitable way of capturing declarative knowledge rather than concrete procedures, and are better connected with natural language.



* The **syntax** defines a set of valid formulas, which are things which are grammatical to say in the language.
* **Semantics** usually doesn't receive much attention if you have a casual exposure to logic, but this is really the important piece that makes logic rigorous. Formally, semantics specifies the meaning of a formula, which in our setting is a set of configurations of the world in which the formula holds. This is what we care about in the end.
* But in order to get there, it's helpful to operate directly on the syntax using a set of **inference rules**. For example, if I tell you that it's raining and wet, then you should be able to conclude that it is also raining (obviously) without even explicitly mentioning semantics. Most of the time when people do logic casually, they are really just applying inference rules.