**Logic, Reasoning, and Persuasion, Week 4-2**
Wrap-up on the Logic Machines! | Normative Logic Machines

## 1 | The Reverse Implication Machine

Suppose we know that if I'm going to the party, then my partner will come. If we also know that I'm going to the party, we can use modus ponens to conclude that my partner will come. But what if instead we know that my partner is *not* coming to the party. Then we can reason as follows:

1. *If I'm going to the party, then my partner will come.*
2. *My partner is* not *coming.*
3. *So I must not be going to the party (since if I were, then my partner would come)*

The implication machine takes an "if P then Q" statement and, if P is true, concludes that Q is true. The reverse implication machine takes that same "if P then Q" statement and reverses two things:

1. It changes truth to falsity
2. It switches P and Q.

This means that it takes that "if P then Q" and, if Q is *false*, concludes that P is also false.

Let $P$ and $Q$ be placeholders for statements, and let NOT mean "it is not the case that" or "it is not true that". So NOT [Adrian is in the classroom] is "It is not the case that Adrian is in the classroom." Then the reverse implication machine permits the following inference:

1. NOT $Q$
2. **If** $P$ **then** $Q$ (or: $P$ **only if** $Q$)

$\rightarrow$ NOT $P$

## 2 | The Gluestick Machine

The gluestick machine sticks statements together, making a bigger statement:

1. $P$
2. $Q$
3. $R$

$\vdots$

$\rightarrow$ $P$ **and** $Q$ **and** $R$ **and** …

Most of the time we'll be simply trying to find the seams where stuff was glued together. We want to find the glue so that we break complicated sentences into smaller pieces and analyze the smaller pieces.

## 3 | Normative Logic Machines

Often arguments are "normative". They are about what you *should* or *ought* to do or believe, where this sense can be *moral* or *nonmoral*. Whenever words like "ought" or "should" or "duty" pop up, we might have a normative argument on our hands.

### *3.1 | The Normative Implication Machine*

Consider the following reasoning:

> *You have a duty to cultivate your autonomy.*
> *You cultivate your autonomy only if you write your own papers.*
> *Therefore, you have a duty to write your own papers.*

**Question:** Is this an instance of the implication machine? **Answer:** No. (Why not?)

If we were to straightforwardly implement the implication machine, we would have to have something like the following:

> *You have a duty to cultivate your autonomy.*
> *You* ***have a duty to*** *cultivate your autonomy only if you* ***have a duty to*** *write your own papers.*
> *Therefore, you have a duty to write your own papers.*

But this is a very different argument! The second premise in the new argument relates two things that you have a duty to do, whereas what we originally wanted was something relating two things that you might *actually* do.

When we have normative language, we need another version of the implication machine, which tells us where the normative language is allowed to be.
Let $A$ and $B$ be placeholders for some actions. Then the normative implication machine permits the following inference:

$$\begin{bmatrix}\text{You ought to do } A\\ \text{You do } A \text{ only if you do } B\end{bmatrix} \implies \text{you ought to do } B$$

**Throughout, you can replace "You ought to" with "You should" or "You have a moral duty to", or "You are morally required to".**

### *3.2 | The Normative Reverse Implication Machine*

If we again let $A$ and $B$ be placeholders for some actions, then the analogue for "ought" is as follows (although it may not be completely obvious why this is true):

$$\begin{bmatrix}\text{You ought not do } B\\ \text{If you do } A\text{, then you do } B\end{bmatrix} \implies \text{you ought not do } A$$

## 4 | The Other Logic Machines

- A complete system of logic machines is a system of logic machines where every truth-preserving argument can be represented as some system of logic machines that feed into each other, so that whenever you start out with true premises (where the premises are statements), you get a true conclusion.
- There are different ways to build complete systems of logic machines. If you start with the implication (and reverse implication) machine, the chain machine, and the gluestick, then you can "complete the system" by adding four more rules: either-or, process of elimination, forgetting, and redundancy. Since these rules are there only for completeness and don't appear often in everyday reasoning, I won't be going through them in much detail.

### *4.1 | The Either-Or Machine*

*We have to either take the Rutgers bus or call an Uber. If we take the Rutgers bus, we'll be late. If we call an Uber, we'll still be late. So either way, we'll be late.*

In this sort of familiar reasoning, we take one "either/or' statement and we derive the same conclusion from both possibilities.

$$\begin{array}{ll} P & \text{if } P \text{ then } Q \\ \text{or} & \qquad\qquad \Longrightarrow \quad Q \\ R & \text{if } R \text{ then } Q \end{array}$$

In general, let $P, Q, R$ be placeholders for statements. Then the either-or machine permits the following inference:

$$\begin{bmatrix} \text{either } P \text{ or } R, \text{ or both} \\ \text{if } P, \text{ then } Q \\ \text{if } R, \text{ then } Q \end{bmatrix} \Longrightarrow Q \tag{1}$$

### *4.2 | The Process of Elimination Machine*

The Process of Elimination Machine works like answering a multiple-choice question via process of elimination. If in a group of statements, we know that *at least one* is true, then if we've determined for all but one that they are false, then we can conclude that the remaining one is true. This is simple: you all do it in multiple-choice exams. If we know that at least one of Emily, Fernanda, George, and Harry signed up for the union, then we can reason as follows:

1. *Emily didn't sign up for the union*
2. *George didn't sign up for the union*
3. *Fernanda didn't sign up for the union.*
4. *Therefore, Harry must have signed up for the union.*

Here's the schematic for two choices: Let $P$ and $Q$ be placeholders for statements.

$$\begin{bmatrix} \text{either } P \text{ or } Q, \text{ or both} \\ \text{NOT } P \end{bmatrix} \Longrightarrow Q \tag{2}$$

### *4.3* | *The Forgetting Machine*

Arjun told you yesterday that he likes cheesecake and chocolate cake. Today Priya asks you whether Arjun likes chocolate cake. Obviously yes: you know from yesterday that Arjun likes chocolate cake. How do we know? Well, Arjun likes cheesecake and chocolate cake. So for the purposes of Priya's question, we can forget about the cheesecake and just remember the chocolate cake part:

*Arjun likes cheesecake and chocolate cake.*
*So Arjun likes chocolate cake.*

This is what the forgetting machine does. In general, let $P$ and $Q$ be placeholders for statements. Then the forgetting machine allows the following inferences:

$$[P \text{ and } Q] \implies Q \qquad\qquad [P \text{ and } Q] \implies P \tag{3}$$

Like the redundancy machine below, this is one of the machines that gives us a "complete set" of machines. But we won't use it much in argument analysis because it does not tend to be used explicitly.

### *4.4* | *The Redundancy Machine*

The Redundancy Machine takes a statement, and then adds *any other statement* and says that one of the two statements is true. For example:

*Ani is at the party.*
***Therefore**, either Ani is at the party or Freya is at the party, or both.*

To be honest, you won't see this one much, since it's usually used implicitly. You probably don't have to remember it.

In general, let $P$ and $Q$ be placeholders for statements. Then the redundancy machine allows the following inferences:

$$[P] \implies \text{either } P \text{ or } Q, \text{ or both} \tag{4}$$

**Exercise:** come up with a real-world use case in which the redundancy machine is relevant. Do the same for the forgetting machine.