Thus, for any nondeterministic Turing machine $M$ that runs in some polynomial time $p(n)$, we can devise an algorithm that takes an input $\omega$ of length $n$ and produces $E_{\omega}$, the running time is $O(p^2(n))$ on a multitape deterministic Turing machine and...

WTF, man. I just wanted to learn how to program video games.
Ontology: knowledge relations

Last time we focused on defining relationships between different pieces of information.

For example, if we know “frogs hop” and “frogs are amphibians”, we can conclude “some amphibians hop”.

Deducing new facts are fundamental to having an expressive knowledge base, as it would be too hard to encode every single fact.
Mental models

However, not all facts are transferable

Consider this information: “I know someone in Italy” and “My friend knows the weather”, but I cannot conclude that “I know the weather in Italy” unless my friend tells me it

Here this is less a physical world fact and more a fact inside my head, which is essentially unknown to anyone else
Mental models

A common way to frame ways of thinking are “Belief, Desire and Intention” (BDI)

https://www.youtube.com/watch?v=96_RSlx2jL0

Each agent has their own local knowledge and goals, which can be communicated
Mental models

Full mental models are an active part of research, so we will focus on just “knows”

$K_a(P)$ will denote agent “$a$” knows fact “$P$”

For example, $K_{James}(\text{next slide})$ as I am aware what the next slide is

We will denote this as $K_J(N)$ for short
Mental models

While I know about myself, I do not know if you know what the next slide is.

However, I do know that you either “know the next slide” or “don't know...”

Thus, $K_J(K_{You}(N) \text{ or } K_{You}(\neg N))$

A world is a possible state of the world that I can be in (i.e. possible cases)
Mental models

A world/case is accessible from another world, if the knowledge of a person is consistent.

In our example with “N” = you know the next slide, we can make a graph:

\[ w_0 : N \quad \leftrightarrow \quad w_1 : \neg N \]

 homosexility accessibility

 = my accessibility

 = your accessibility
Mental models

I have a link between $w_0$ and $w_1$, as in both worlds $[K_{\text{You}}(N) \text{ or } K_{\text{You}}(\neg N)]$ is true.

You however know whether or not you know the next slide (e.g. $K_{\text{You}}(N)$), so you cannot go between these two worlds.

Both possible worlds exist, as in the model I am unsure (despite you knowing).
Mental models

Let's model another fact: Wednesday's topic (denoted “W”), which only I know... graph is:

\[ w_0 : N, W \]
\[ w_1 : \neg N, W \]
\[ w_2 : N, \neg W \]
\[ w_3 : \neg N, \neg W \]

\[ \longleftrightarrow \text{ = my accessibility (no self arrows)} \]
\[ \text{red } \longleftrightarrow \text{ = your accessibility (no self arrows)} \]
Mental models

You try it! What if I did not know Wednesday's topic either? How would this change?

\[ w_0 : N, W \] \hspace{1cm} \[ w_1 : \neg N, W \]

\[ w_2 : N, \neg W \] \hspace{1cm} \[ w_3 : \neg N, \neg W \]

\[ \Leftarrow \Rightarrow \] = my accessibility (no self arrows)

\[ \Leftarrow \] = your accessibility (no self arrows)
Mental models

You try it! What if I did not know Wednesday's topic either? How would this change?

\[ w_0 : N, W \]
\[ w_1 : \neg N, W \]
\[ w_2 : N, \neg W \]
\[ w_3 : \neg N, \neg W \]

\[ \longrightarrow \text{ = my accessibility (no self arrows)} \]
\[ \longrightarrow \text{ = your accessibility (no self arrows)} \]
Mental models

We can actually combine them:

\[ w_0 : N, W \]
\[ w_1 : \neg N, W \]
\[ w_4 : N, W \]
\[ w_5 : \neg N, W \]
\[ w_6 : N, \neg W \]
\[ w_7 : \neg N, \neg W \]
\[ w_2 : N, \neg W \]
\[ w_3 : \neg N, \neg W \]

\[ w_0 \text{ to } w_3 = \text{I do know } W, \quad w_4 \text{ to } w_7 = \text{I don't} \]
Mental models: logic

Logic rules apply to this “knows” as well

For example, if Bird(x) => Fly(x)
Then, $K_J(Bird(tweety)) => K_J(Fly(tweety))$

This can extend to mental implication as well (for facts that only one entity knows):
($K_J(P) ^ K_J(P => Q) => K_J(Q)$)
Mental models: logic

However, you have to be careful with $K_a$, as the order matters with previous logic ops.

For example:

$\exists x \ K_a(Friend(x)) = \text{in every possible world, you have someone who is your friend}$

$K_a(\exists x \ Friend(x)) = \text{you have a friend in every world, but are different people}$
You must also put $K_A$ before any knowledge piece that is specific to a person.

$$K_A(P \text{ or } \neg P) \equiv K_A(\text{True}),$$ which is a worthless statement ("A knows true things are true")

$$[K_A(P) \text{ or } K_A(\neg P)]$$ is a useful statement, which indicates that "A knows the state of $P$."
Mental models: inference

With the additional rules for $K_a$, you can use ordinary first-order logic to resolve statements.

However, this might not be enough... Consider this picture:

It is just some trees, right?  ... Right?!
Mental models: inference

Quite often humans make assumptions, even if we do not have any exact evidence.

Suppose you go outside and it is wet, you might assume that it rained.

However, you have no actual information about this... maybe there was a squirt gun fight.

Yet, this assumptive logic can be powerful.
Mental models: inference

Two assumptive strategies are:
1. Circumscription - Assume any unsaid statements are false (sound familiar?)
2. Default logic - Add assumptive logic rules are added unless they contradict current known rules

Both of these require writing more logic statements: either (1) modifying existing ones or (2) new prospective ones
Circumscription can handle exceptions so that you can write generalizations

\[ \text{Dolphin}(x) \Rightarrow \text{Blue}(x) \]

This is useful, until you see this:

To handle this, you need to add an exception (that you assume is false):

\[ \text{Dolphin}(x) \land \neg \text{Exception}_0(x) \Rightarrow \text{Blue}(x) \]
Default logic follows the same general format:

\[ P : J_1, J_2, \ldots J_n / C \]

Here P is a precondition, J are justifications, and C is the conclusions...

We can write our pink dolphin as:

\[ \text{Dolphin}(x) : \text{Blue}(x) / \text{Blue}(x) \]

This says: “If x is a dolphin, assume it is blue, unless being blue causes a contradiction”
Default logic follows the same general format:

\[ P : J_1, J_2, \ldots J_n / C \]

Here P is a precondition, J are justifications, and C is the conclusions...

We can write our pink dolphin as:

\[ \text{Dolphin}(x) : \text{Blue}(x) / \text{Blue}(x) \]

This says: “If x is a dolphin, assume it is blue, unless being blue causes a contradiction”
Fixing: assumptive logic

If we used forward chaining to generate a large amounts of facts from a database...

When a contradiction happens, we have to re-generate part of the knowledge base

We cannot simply negate one step, as we might have built other inferences on top of one assumption
Fixing: assumptive logic

Consider this KB: \[ P \Rightarrow Q \]
\[ Q \Rightarrow S \]
If we assumed P, we then conclude: \( Q \land S \)
If P caused a contradiction later, we would have to remove both Q and S

But if the KB was:
\[ P \lor R \Rightarrow Q \]
\[ Q \Rightarrow S \]
\[ R \]
Now even if P is a contradiction, we want to keep both Q and S
Fixing: assumptive logic

This reformulating of the knowledge base is crucial for large knowledge bases.

There are efficient strategies to save inference deductions so facts can be quickly flipped on and off given the current knowledge base.

For example, you can store the various reasons that make a fact true... and if all of them are contradictions it should be removed.