Computers and Logic

Hard Problems for Computers

<u>Not all problems can be translated into a problem space.</u> Some problems are difficult in interesting ways both for people and for machines.

Examples:

1. **Topological Puzzle**

   They come in various forms: rings, or wires, or things like that. In general, the structure of these puzzles is that <u>you have to take them, they're physical objects, and then you have to manipulate them in some way to separate pieces or reconnect pieces.</u>

   It really seems that when you look at puzzles like this, the difficulty, <u>the essential difficulty is finding out what the states are, what constitute the meaningful states of the puzzle,</u> the positions of the rings, and usually from that point the moves are not so hard if you know what the states are.

2. **Math book problems**

   Here’s a problem from How to Solve It: Modern Heuristics

   > *Mr. Smith and his wife invited four other couples for a party. When everyone arrived, some of the people in the room shook hands with some of the others. Of course, nobody shook hands with their spouse and nobody shook hands with the same person twice. After that, Mr. Smith asked everyone how many times they shake someone's hand. He received different answers from everybody.*
   >
   > *The question is: how many times did Mrs. Smith shakes someone's hand?*

   The kind of techniques that we would use to solve such puzzles are described in **How to solve it by George Polya**.

   Polya popularized the term **heuristic**, meaning <u>a rule of thumb, a strategy to try in solving a puzzle or a problem</u>. Not guaranteed to solve the puzzle, but often helpful, often the thing that you should have in your bag of tricks, that you should have in your repertoire.

   A certain heuristic which may be useful in solving the above problem is to doodle out a figure. Polya mentions some other typical strategies of proofs like **reductio ad absurdum** or **induction** or **decomposing the problem into subproblems**, **dimensional analysis** is a very helpful technique for solving physics problems where you can just reason about the dimensions of the quantities involved.

3. Physics problem

> *Imagine two quarters side-by-side. Now you take the quarter at the right, the one which has the head side up, and without slipping, carefully you roll it around so beside the quarter with the tail side up. The tail side up quarter remains stationary the whole time. So once you've rolled this head side of a quarter to the opposite, to the left side of the picture, which way will George Washington be facing?*

The way that we tend to solve questions like this is through **dynamic mental imagery**. Mental imagery that executes animation. That's a very different programming than the programming that would be involved in trying to represent this as a problem space with moves. To solve a problem like this, you would want to get a computer to be able to take sample animations or sample pictures and animate them in realistic ways to get an answer.

> *Imagine that you have a frictionless track, a ping-pong ball, it's rolling to the left at one centimeter per second and a bowling ball is rolling to the right at once centimeter per second, and they're going to collide and then after the collision, they will have new velocities. So a totally elastic collision. The ping-pong ball is one gram in mass, and the bowling ball is 10,000 grams in mass. You need to solve for the final velocity of both the ping-pong ball and the bowling ball if you use the fact that energy is conserved and overall momentum is conserved.*

This is another example where we cannot specify the search space. We can solve this problem by using relative motion.

4. William Pounderstone's Labyrinth of Reasons

> *A man gets an unsigned letter telling him to go to the local graveyard at midnight. He does not generally pay attention to such things, but compiles out of curiosity. It is a deathly still night, lighted by a thin crescent moon. The man stations himself in front of his family's ancestral crypt. The man is about to leave when he hears scraping footsteps. He yells out, but no one answers. The next morning, the caretaker finds the man dead in front of the crypt, a hideous grin on his face.*
>
> *The question is, did the man vote for Teddy Roosevelt in the 1904 US presidential election?*

Unlike the kinds of problems that Polya talks about, there is a fair amount of extraneous data here. There's stuff that is there to not exactly to lead you astray but stuff within the story that is not relevant to the solution of the problem. There's a little bit that is relevant to the solution of the problem. Even then, solving the problem requires a certain amount of background knowledge of a kind that not everybody has.

Machines and Logic

One of the more pointed comparisons between the ways that people think and the way that machines think, has to do with debates and discussions around the theme of **logic**. Logic is a very broad term. Logic in general actually refers to a tradition of thinking that goes back to at least as far as Aristotle who wrote about logic. Aristotle's portrait of logic centered around **syllogisms**, patterns of reasoning like,

> *All men are mortal. Socrates is a man. Therefore, Socrates is mortal.*

The discussion having to do with machines and minds often centers on this idea that machines are particularly logical or the ways in which computers can think is especially logical.

George Boole's The Laws of Thought:

Boole's book he mixes different formalisms from things that today would be referred to as **Boolean logic**, that is logic having to do with ones and zeros. **Propositional logic**, that is logic based on the idea of manipulating true or false sentences. Finally, **set theory**.

Proposition Logic:
In propositional logic, the idea is that letters or symbols stand for entire propositions that can be true or false. Now, propositional logic allows you to reason with sentences of that kind by combining them with what are called connectives, like **and** and **or**, **if then** and then doing certain fairly straightforward reasoning using the propositions and sentences that you've asserted.

You have a bunch of assertions. You have a bunch of sentences that you say are true, we're going to treat as true.Then from those sentences, we will see what else we can deduce that should also be true.

Assertions:

1. *IF P OR (NOT Q) THEN R*
2. *Q*
3. *P*

Deductions:

1. *P OR (NOT Q)* *from 3*
2. *R* *from 1 and 4*

Computers are really good at this. So propositional logic does lend itself well to programmed implementation.

Standard deductive steps in Propositional Logic:

The discussions around logic as applied to people are often an uneasy mix of descriptive and normative. *Sometimes people want to argue that we do think logically and sometimes people want to argue that,even if we don't, we should.*

It certainly lends itself well to machine reasoning in many cases, not in all. But it often lends itself well to machine reasoning because it follows sets of formal rules, that for people can often be stressful to follow in ways that are free of mistakes. We often make mistakes. Machines when suitably programmed can be very effective at doing this.

1. **Modus Ponens**

   In propositional logic a number of these standard deductive steps are called modus ponens. It's a very old rule in logic. Basically, it says that if you know that if A, then B is true, if you know that that's true and you know that A is true, then you know that B is true.

   If there is fire, there is smoke.
   There is fire.
   Therefore, there is smoke.

2. **Modus Tollens**

   An equivalent rule that is to say equivalent to modus ponens means really the very same thing, but it goes by a different name modus tollens. It has a different syntactic structure where we can say, if there is fire, there is smoke and we know there is not smoke,therefore, there is not fire. Because if there were fire, there would be smoke.

   If there is fire, there is smoke.
   There is not smoke
   Therefore, there is no fire.

Modus ponens and modus tollens are both totally perfect deductions from two earlier sentences. They only have a slightly different, syntactic structure. For a machine, these two things are essentially the same. For a person, they're not quite the same. People find it much easier to reason using modus ponens than they do using modus tollens.

Logical Fallacies:

The ways in which typical computer programs deal with propositional logic and the ways in which people often informally deal with logic already have some differences to them. People also are prone to make mistakes in this logic.

1. **Denial of the Antecedent**

   If there is fire, there is smoke
   There is not fire
   There may or may not be smoke. (there might be smoked for other reasons than fire)

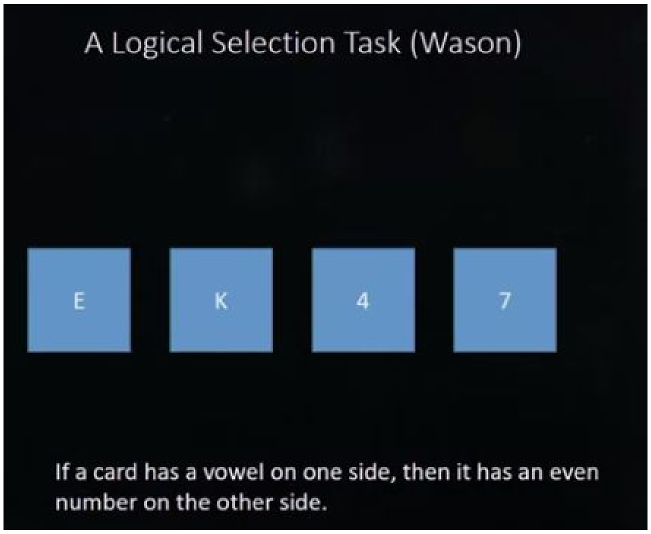
2. **Affirmation of the Consequent**

   If there is fire, there is smoke
   There is smoke
   There may or may not be fire.

The fact that we are prone to make certain logical mistakes, tells us that there's more to human reasoning than is expressed in the formal rules of logic. Propositional logic doesn't seem to be the laws of thought. It seems to be an element of a formal representation of certain kinds of effortful thought, but we don't follow the rules of propositional logic terribly faithfully.

Example:

> *People are told that they're given four cards. They are told that on one side of the card is a letter and on the other side of the card is a number. Then you are given a rule which your job is to test. You want to test this rule to see whether it's true or false. So the rule that you're going to be given to test is, if a card has a vowel on one side, then it has an even number on the other side, and you are asked, given this set of four cards to turn over exactly and only those cards that you would need to test whether this rule is true.*



You need to turn over exactly two cards -

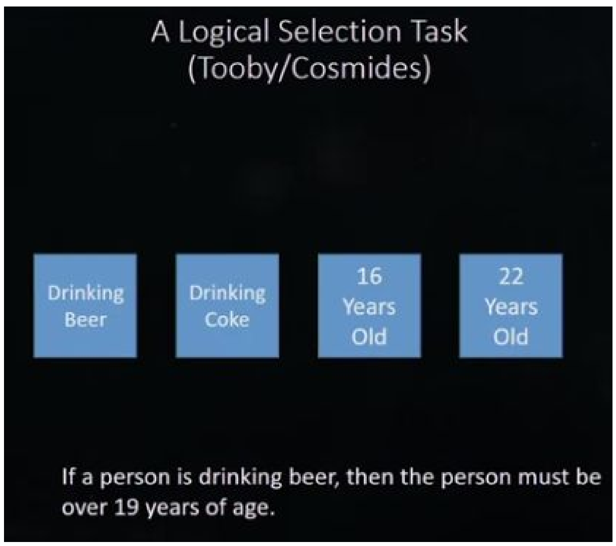
1. You need to turn over the E. If the card has a vowel on one side, then it has an even number on the other side. If you turn over the E and you find an odd number, you know the rule is false.

2. The other card that you have to turn over is the seven. If you turn over the seven and you find a vowel like an A, then you know the rule is false.

Most scientists don't feel that 's an accurate representation of scientific pursuit. But nonetheless, some people argue that when you do an experiment, it should be with an eye toward disproving a theory.

A similar example is given where,

> *You're given four cards. On one side of the card is the drink that a person is having. On the other side of the card is the person's age. Your job is to see whether this rule is true, or if this law is being upheld. If a person is drinking beer, then the person must be over 19 years of age. Your job is to turn over only and exactly those cards that will show whether the rule is being held to.*



You need to turn over exactly two cards -

1. You turn over the drinking beer card. If the age on the other side is 16, then the rule is being violated.
2. You turn over the 16-year-old card, if the person is drinking beer the rule is being violated.

The other two cards, you don't need to turn over, regards because for example, turning over the drinking coke card can tell you nothing about whether the rule is being violated, similarly with the 22-year-old person.

In this case, people have a much easier time solving the problem.

So why is it that this task seems to be so much easier than the previous task involving letters and numbers?

There are different explanations for this. There's not a unique explanation to it. The original experimenters who presented this version of the card task would make an argument that goes roughly as follows: we are very good reasoners when it comes to situations that are ecologically or evolutionarily realistic for us. but seeing whether laws are being followed, seeing whether rules are being obeyed or violated, that is a very, it's a venerable human activity.

The ways in which people reason in situations that could be modeled logically doesn't seem to be quite the same as the ways in which the rules, the formal rules of logic dictate or pure mathematics dictates or in fact, the ways in which it is relatively easy to program machines. Machines can be made to follow logic reliably. For us, it seems to be more of a problem.