# Mechanical Intuition

Lumberjacks Incorperated (2018)

## Preface: Algorithms, Problems, and Solutions

We formalise an algorithm the same way most modern definitions do. We say an algorithm is a computable function, that maps an input (which contains a representation of the problem, as well as other information needed) to an output (which is the solution to the problem).

| Input | -> | Algorithm | -> | Output |
|---|---|---|---|---|
| (Problem Instance + Information) | | (Computable Function) | | (Solution) |

On the computable transformation however, we loosen the definition to not include termination, and only require that if the computation terminates, it has terminated with an output with the correct solution.

## Two Different Types Of Problem Solving Machines

Let us without context, consider two different types of machines that run algorithms for solving problems.

*Type 1 - Finite Input Machine*

One machine takes a finite and definite input given at the beginning of the machine computation, that represents the problem instance, and the machine performs a computational transformation on the input, so that if the computation terminates, it terminates with the correct solution to the problem instance.

*Type 2 - Continous Input Machine*

The other machine takes a finite and definite input at the beginning of the machine computation, that represents the problem instance, and as well as this is fed new input continuously forever or until termination (distinct), such that if the computation terminates, it terminates with the correct solution to the problem instance.

Thus, we have one machine that is given all of the information required for the computation at the beginning of the computation, and one that is continuously fed distinct input, forever, or until termination.

## Solving Problems and Information

When we use an algorithm to solve a problem, we have a computational transformation from input to output, where the input is both a representation of the problem, plus other information needed by the computation to 'solve the problem', or transform the problem representation into a solution representation.

To break this down further, we have two options:
(both involve needed a finite amount of information to solve the problem)

*Option 1: Full Finite Definite Information Problem Solving*

What we mean by this, is not only a representation of the problem, but also the finite extra information required is known exactly, so that all can be entered at the time the computation begins. This is another way of saying the information needed to solve the problem is exactly known.

*Option 2: Indefinite Information Problem Solving*

What we mean by this, is that although we have a representation of the problem, we do not know what the exact information needed to solve the problem. How we solve a problem without knowing the exact information is by entering random information that has a probability of being correct, and continue to enter data of this type to the machine.

If we continuously enter distinct data, and the data required is finite, we always have a probability that at some point we enter the correct information needed, and the algorithm will terminate with the correct solution.

Using this approach can be useful, as it allows us to solve a problem instance without knowing prior what exact information is needed, although this may never terminate if the correct data is never reached (i.e if possible input are infinite), however, theoretically, in this problem solving type, at time infinity the solution would be reached.

## Problem Machines and Problem Solving

'Full definite information problem solving' is best suited to 'Type 1' machines. Why? If we know the exact input needed for solving the problem by the algorithm, then we can use this with a type 1 machine, and enter the finite exact data at the start of the computation.

This type can however also be used with machines of 'Type 2', it just means that information that flows in is not needed, or redundant.

'Indefinite information problem solving' cannot be done on machines of 'Type 1', since a continuous information stream of information that may not be finite is needed for the computation. It is however, suited to machines of 'Type 2'. Since we do not know the exact information needed to solve the problem, we use the property that machine 'Type 2' provides a continuous and distinct stream of data to achieve this.

Thus, machines of 'Type 2' can solve problems where the exact information needed to solve the problem is not prior known.

## Intuition

What do we mean when we talk about intuition with regards to problem solving?

Often when we talk about problem solving, we talk about reaching the solution in a number of steps. When we take a step, we can either take a reasoned step that we know with certainty will lead to the solution, or take a step that has a probability less than certain of reaching the solution.

We may take the 'intuitive' step approach when we do not have enough information to determine the certain step. We hope that we eventually reach the correct solution, and although we may take the wrong step, as long as we keep attempting a solution, gaining information about what is a wrong step, then we will approach the solution as the number of steps approaches infinity. We have a probability of reaching the solution after the finite number of steps, and theoretically, if a solution exists, we would always reach the solution at infinite steps.

## Solving Using Intuition on a Machine

*How does intuition translate to a computations?*

We can think of using intuition to solve a problem as a form of 'Indefinite information problem solving'. In both, we do not have the required information to solve a problem, and we make 'guesses'. We can represent these continuous 'guesses' as the continuous stream of information used in 'indefinite information problem solving'.

Hence, we can represent the intuitive approach to solving a problem with 'indefinite information problem solving'.

*How does intuition translate to a machine?*

We already know that we can use machines of 'Type 2' for the 'indefinite information problem solving' problem, and we extend this using the point above, to arrive at the conclusion that these machines can be used to implement the 'intuitive' approach to problem solving.

The intuitive approach can be taken in situations where the exact information needed to solve a problem is not prior known, and implemented on continuous information stream machines ('Type 2' machines), but the tradeoff is that we may never arrive at the correct information needed to solve the problem (Only at time infinity can we be sure to have tried all possible information in all situations)

## Tuning Intuition

For performance purposes, when considering the intuitive approach, we want to reach the solution as fast as possible.

To do this, we want to make the best 'guesses' for the next data the that is inputed in the continuous stream to the machine.

We say that there is 'stronger intuition' if the guess has a higher probability of being correct.