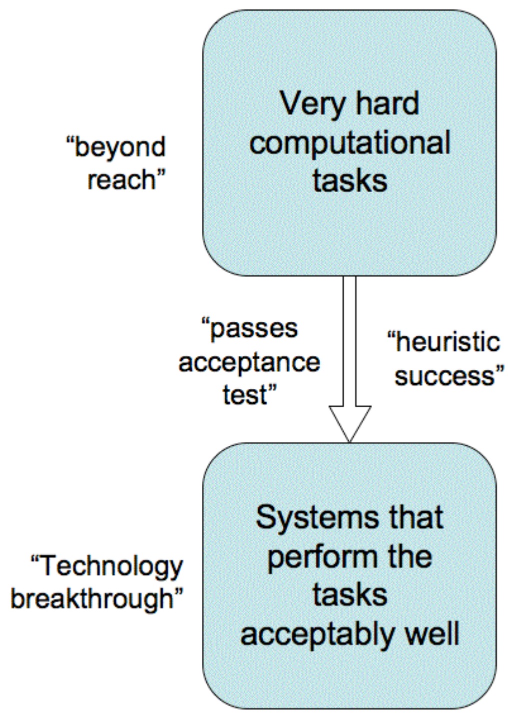
Automation Principles

8/15/07

These principles concern finding efficient computational ways to perform human tasks. Tasks can be physical, such as running an assembly line, driving a car, controlling airplane surfaces; or mental, such as doing arithmetic, playing chess, and planning schedules.

Physical automation maps hard computational tasks to physical systems that perform them acceptably well.

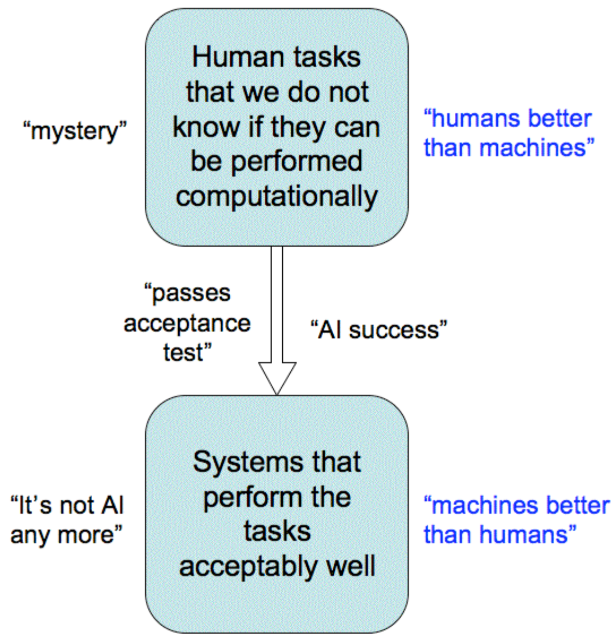


A. The class of very hard computational tasks includes most of the problems people want to solve for business, science, and engineering; for example, figuring route capacities in a transportation network or numerically simulating an aircraft in flight. These problems have all been proved to be NP-hard or worse.

B. There is tremendous motivation to find approximate or heuristic ways to solve them, not perfectly but acceptably well. The arrow in the diagram above means that very hard tasks are mapped to simpler systems that do good-enough jobs. This is a form of automation because what was previously beyond reach can now be done well by a machine. It is not unusual for a heuristic system that does consistently well with a (formerly) computationally hard task to be celebrated as a technology breakthrough.

C. The artificial intelligence branch of the computing field has developed and studied search processes that reduce very hard tasks to workable ones. For example, a genetic algorithm can generate a set of candidate solutions to the problem and then transform them through several generations of cross-matches and mutations until it evolves a good enough solution.

Artificial intelligence maps human cognitive tasks to physical systems that perform them acceptably well.



A. In much the same way as physical automation, the field of artificial intelligence is concerned with mappings from tasks to good-enough systems. In this case the tasks are cognitive tasks of humans and possibly animals. The big difference is that many human tasks are not known to be computational, and therefore solutions cannot start with the assumption they are trying to simplify hard computational problems.

B. People in the AI field work with two different hypotheses about human tasks. The *strong AI hypothesis* holds that human tasks (second diagram) are ultimately computational; they are a subset of hard computational tasks (first diagram). Strong AI seeks to discover the computational methods and automate them. The *weak AI hypothesis* holds that human tasks may not be computational but that we can nonetheless find good-enough computational equivalents. Weak AI seeks to construct computational systems that mimic human tasks acceptably well.

C. There has been considerable argument over the validity of the strong AI hypothesis. A famous “Chinese Room Argument” by John Searle shows that a standard computer that speaks fluent Chinese would be doing nothing more than manipulating symbols without regard to their meaning and would embody no intelligence; therefore a perception by a Chinese speaker that it is intelligent is ungrounded. In practice, this argument is mostly theoretical because most people won’t accept that a computational system can perform human tasks unless the system passes a battery of acceptance tests. In other words, they employ the weak AI hypothesis to guide their acceptance even if they worked from the strong AI hypothesis to propose a system.

D. Behavioral acceptance tests are used to decide whether a computational system successfully performs human tasks. They compare the observed behavior of the computational system with the observed behavior of humans doing the task. The Turing Test, which compares conversational behavior of humans and machines, was an early example.

E. When there is no working good-enough computational system for a human task, we are likely to say: “It is a mystery how humans do this, and that’s what distinguishes humans from machines. For this task, humans are better than machines.” Once an acceptable computational system has been found, we are likely to say: “This is clearly not an intelligent task because a machine can do it. For this task, machines are better than humans.” In other words, tasks stop looking intelligent when we find computational systems to perform them.

F. There are numerous examples of the tendency to hold a task as intelligent as long as no computational method for the task is known. One of the earliest was the game of chess. Early AI researchers thought that finding machines to play chess would be a great accomplishment in the progress of intelligent machines. Today chess programs running on personal computers play at the master level. However, no one thinks the programs are intelligent or even that skill at chess is an intelligent activity. The methods of modern chess programs -- brute force searches of future boards -- do not resemble how the chess masters think when they play.

G. The tendency to think technology is unintelligent manifests in many fields, not just artificial intelligence. Arthur Clarke famously said: “Any sufficiently advanced technology looks like magic." Users of automation say: "Any magical mystery when sufficiently understood looks like advanced technology."

Artificial intelligence maps tasks to good-enough computational systems through models, search, deduction, induction, and collective intelligence.

A. Modeling means employing (or positing) a model of the intelligent behavior; and then simulating the model with a computer.

B. Many computational tasks are hard because the solution requires the algorithm to aggregate components from very large state spaces. Search uses clever heuristics to reduce states included in the solution to a manageable number.

C. Deduction means to follow the rules of logic to construct true statements that follow from other, already-known, true statements.

D. Induction means to generalize from known statements, data, and evidence to models within which useful deductions can be made.

E. Collective intelligence means to aggregate individual knowledge across a network or a game to produce new knowledge and new actions.

F. In all five, knowledge representation is a major issue. A representation is a mapping from the current situation (often called a frame or frame of reference) to patterns of symbols obeying a syntax; the semantics (meaning or interpretation) of symbol-patterns is constructed by the observer. Humans almost always incorporate tacit, contextual information that they are not aware of. Therefore, the mapping from the original frame to the representation necessarily loses information, and thus the results of a computational system may not agree well with the expectations of the human observers. This potential for disagreement is inherent in all simulations of cognitive tasks. It is called the *frame* or *context problem*.

Models represent processes by which intelligent beings generate their behavior.

A. A model is a representation of a physical system as a set of equations or algorithms that can be used to calculate the system's response to various stimuli. A model of intelligent behavior may be computational even if the methods used by the brain and nervous system are not. A model is validated by comparing its responses to those of the real system and confirming that they are close most of the time.

B. Many in the AI field have proposed theories of how intelligence works and then offered computational models that allow others to study and evaluate the theory. A non-exhaustive list of examples follows.

C. Logic systems model cognitive tasks as the sifting of many facts and previous deductions until a conclusion is reached. Fuzzy systems are a generalization of logic systems that allow *degrees of truth*, expressed as numbers between 0 and 1.

D. Expert systems are based on a model that experts apply rules mapping current sensory input and experiential memories to actions.

E. Neural networks are mathematical models that seek to realize intelligent functions by partially imitating the neuronal connection structures of nervous systems.

F. Sparse distributed memory models consciousness as a stream of states circulating through a large memory system that stores copies of binary patterns in many cells distributed throughout the memory.

Search finds the subsets of states of a complex system that must participate in the final outcome of a task.

A. Many tasks require solution procedures to aggregate components from throughout very large state spaces. For example, the traveling salesman problem, asks for the shortest tour of N cities on a map. A solution procedure enumerates all N! tours, calculates the length of each one, and selects the shortest. Unfortunately, the computation time for this is of order $O(n^n)$, which becomes completely unfeasible even for small numbers of cities.

B. Search is a process of enumerating the states and deciding which ones should participate in the solution. Heuristic searches can be significantly shorter. A heuristic for the traveling salesman problem might generate a random tour and then, for every pair of cities, exchange the visit order for the two cities if that shortens the tour; it then offers the shortest tour found in this $O(n^2)$ process as a good-enough solution. Other examples are search are noted in the next points.

C. Many state spaces can be organized as trees of complex sub states. Each descendant of a node represents a possible future sub state under a particular decision by the system when in that node. The search process can reduce the size of the tree by ruling out descendants that are unlikely or have low value to the outcome. A good example is chess. A board configuration is a sub state. A descendant is another board reached by a single move. A chess-playing machine tries to enumerate these board states, looking for move sequences that lead to a win. Heuristics that limit the tree include ruling out boards that result from moves that lose too many "piece points" and boards that are more than a few moves in the future. The depth of the tree that a machine can explore in this fashion depends on the speed of its processors and the size of its memory.

D. Genetic algorithms are a form of search that mimics the way cells divide and recombine DNA in living organisms. A solution to the task is represented as a binary string, and each string is assigned a "fitness" number. The higher the fitness, the better the solution represented by a string. The genetic algorithm runs through "generations", which are sets of N strings representing solutions. It calculates the fitness of each string and discards the N/2 least fit. It generates replacements for those discarded by picking pairs of most-fit strings with probabilities proportional to their fitness, and recombining them by exchanging prefixes relative to a random crossover point. It also occasionally reverses a bit of one of these new strings (mutations). After a few generations a string of dominant fitness will emerge and will often have fitness very close to the optimal fitness that would be found by enumerating all possible solution strings.

E. The Internet search engine Google builds a database of summaries of web pages that it visits by following links from other web pages it has already visited. It ranks each page according to the number of links pointing to it weighted by the ranks of the pages containing those links. When a user presents keywords, Google returns the list of summaries in decreasing order of rank. Most of the time, users find at least one useful answer among the top ten proposed by Google.

F. Neural networks are often used for searches. A network is trained by presenting it with a set of (X,Y) pairs; X represents a sensory input and Y the desired response. The network's internal parameters are adjusted so that for every X in the training set the network produces the correct Y. Once this has been done, the Y values can be found very quickly by presenting their X's. And new Y values can be estimated by presenting new X values (outside the training set).

Deduction locates the outcome of a task by applying rules of logic to move from axioms to provable statements.

A. A logic system consists of a set of axioms and rules of inference. A rule of inference specifies a new proposition that can be formed from axioms and other propositions. A proof is a sequence of statements each of which is either an axiom or a proposition formed from applying an inference rule to a subset of the previous statements. Logic systems are powerful models for processes of deduction. Alan Turing believed that deductive ability is the core of problem solving and is the mark of intelligence; he stressed a program of research in AI that emphasized deductive systems.

B. Expert systems are a major application of logic system. They represent expert behavior as large database of facts and inference rules. A machine takes a description of a current situation and seeks the deductions the expert would make from the same experience and facts. Expert systems have been most successful in diagnosis, where one proceeds from symptoms to a conclusion about the problem causing the symptoms, which leads in turn to a treatment. Examples of diagnosis include medical problems, mechanical problems in cars and airplanes, and recommendations for computer system configurations given the needs of a user. Expert systems have not been successful when the expert behavior includes creative generalizations that cannot be deduced from any existing rule.

C. The language Prolog is specifically designed for logical deduction. It derives the answers to queries by searching through rules and facts, looking for all valid deductions. The Japanese Fifth Generation Computing Project (1980s) envisioned a powerful logic supercomputer for Prolog that could process billions of logical deductions per second.

D. Another major application of logic systems is in formal proofs of programs. Being able to prove that a program works correctly was once considered a human task ripe for automation. The input and output of a program are described by logic statements. Each sequential, conditional, and iterative structure of the program transforms its local input logic statement to a local output logic statement. The prover tries to construct a derivation of the program output logic statement from the program input logic statement.

E. For many years many people believed logic systems would explain most intelligent behavior. They reasoned that a brain is a very complex state system and that a sufficiently complex logic system could capture all its firing rules and allow the logic system to reach precisely the same deductions as the brain. No one ever achieved this goal. The last and only project pursuing it is Cyc (cyc.com), which is attempting to see if a database of trillions of facts and rules could allow the computer to reach human-like conclusions all the time.

Induction builds models by generalizing from data about a complex task's behavior.

A. The human brain is capable of induction, the construction of a model based on available data. For example, someone hypothesizes that the moon moves in an orbit at a constant speed and direction relative to the earth's surface. From the data about moon positions for several consecutive days, the person can then use the model to calculate the moon position on future days.

B. All the cases of inference can be stated in a simple general form. We are given a set of data D observed in the world. We would like to form a hypothesis H that explains the data and allows us to predict new data values in the future. Typically the data are in the form of pairs (X,Y), where X is a stimulus to the system and Y is its response. Since the human brain has the capability to form generalizations and abstractions, methods for induction would help computers accomplish some cognitive tasks. Three approaches to this are curve fitting, Bayesian inference, and reinforcement feedback.

C. Curve fitting is a class of induction methods that find mathematical curves that minimize error relative to the data points. A linear curve fitter, for example, assumes that the responses of a system (Y) are a linear function of its stimuli (X). Given a set of data in the form of samples (X,Y), the linear curve fitter finds a constant A such that the total of the errors $(Y-AX)^2$ is smallest. The curve fitter thus creates its best estimate of the relationship Y = AX. Then in the future, someone who wants to know how the original system would respond to a new input Z, would estimate that response as AZ. This situation resembles the training of a neural network mentioned earlier, except that the neural network is simply trying to mimic the original "X produces Y" behavior without trying to fit the data to a specific function.

D. Bayesian inference is another class of inference functions. It seeks to construct the most likely hypothesis H given the observed data D. As part of choosing H, we want to evaluate p(H|D), the probability that H is true given the data. Bayes's theorem says that p(H|D) = p(D|H)p(H)/p(D). It is usually easy to calculate the probability that the hypothesis is true in general, the probability the data will be observed, and the probability that the data will be observed given the hypothesis. The Bayesian inference engine can operate in several modes. One is that it is given as input several hypotheses, and it simply calculates which one is most probable. Another is that it has a way to iteratively improve a hypothesis (e.g., choosing the parameter A in a hypothesis that Y = AX) and it does so until it can find no further improvement.

E. Reinforcement feedback is another class of inference functions. The objective is to adjust internal parameters of a system so that its responses are as close as possible to the desired responses. The desired responses can be represented as a training set of data points (X,Y), where X is an input and Y the desired output. After each input X, a learning algorithm adjusts the internal parameters to get the system to produce an output as close as possible to the desired Y. When all the points have been presented, the system has undergone multiple adjustments and gives reasonable performance for all the data points in the training set. The adjustment algorithm is a way to represent how the system learns from its data. Adjustment algorithms can be run once for an initial training set, or run repeatedly as new data are encountered in experience.

Collective intelligence exploits large scale aggregation and coordination in networks to produce new knowledge.

A. The average of numbers contributed by many individuals is often a more accurate predictor than individual guesses. Some companies operate internal virtual stock markets in which individuals buy and sell stocks representing future products; average stock prices are often excellent predictors of actual product performance in the real market.

B. In a search engine, each search contributes to the database of history of all searches. When combined with that historical information, a user's keywords can be used to infer what the user actually intends in a search, giving a more accurate response.

C. Crowdsourcing is a new way to combine the knowledge of a community into solutions for challenge problems. The challenger advertises a problem; then other community members annotate it with possible solution approaches. The challenger can select the best one and help that person achieve it.

D. Human computation games are multiplayer games in which, as a byproduct of the play, the players apply human intelligence to solve problems that no known machine can. For example, in the game esp.com, pairs of players cooperate in the labeling of images; the labeled images can then be found by search engine video searches.