Lecture 15. The Limits of Computing and Artificial Intelligence

Note: in cases I were unable to find the primary source of an image or determine whether or not an image is copyrighted, I have specified the source as "unknown". I will provide full information about images and/or obtain reproduction rights when such information is available to me.

Introduction

We are living in digital society, submerged in digital culture. It is difficult to imagine our lives without computers, without infrastructure that they provide. If the history (i.e. the recorded past) is any guide–and we clearly don't have anything else as powerful as our past experience–any major technological breakthrough or scientific discovery inspires Utopian sentiments. In the past, the advent of steam power, electricity, electronics, and even the Internet, had inspired ungrounded proclamations of a techno-panacea leading to ever lasting technology-based higher forms of social organization (and liberation). The world would be better, freer, more fair as long as steam power is with us – we proclaimed a few centuries ago. But we know that steam power was only a stage in the development of our civilization. It kick-started the Industrial Revolution, modernized almost every sector of manufacturing and produced repeated immense waves of economic growth. It made a lot of people rich, fast, and left many people poor and desperate. And then it was over.

Will computers be always with us? Will they make us better, freer, more fair? Are they the ultimate tools that, when they evolve sufficiently, will be able to provide us with limitless benefits, and guide us to the next and higher step of social and cultural evolution? Or will they share the fate of steam engines?

Perhaps computers will evolve into some other devices, like calculators which seemed invincible but, in the end, only paved the way to computers. Perhaps in 50 years computers might only be seen in technology museums...

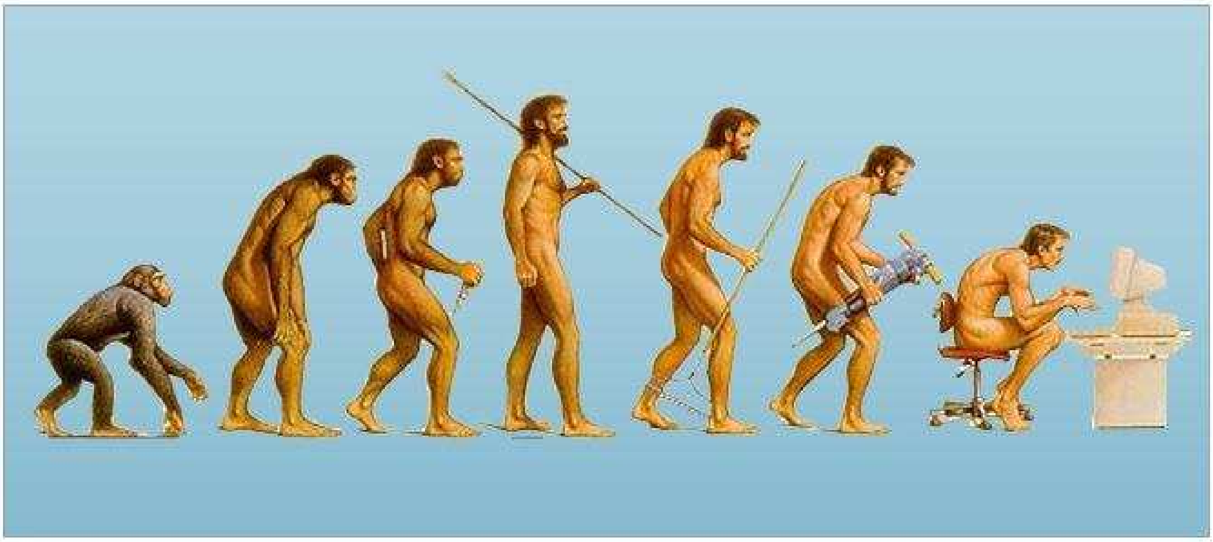

Fig. 1. The evolution of computing. Source: unknown.

To search for answers to some of these questions one may consult computer scientists themselves. What do they know about computers and computing? Will the continuous technological progress result in future computers that will be able to solve all problems of interest to us? What does computer science know about the limits of computing?

In this lecture we shall search for scientific answers to just two questions. The first deals with the limits of computing. The second with the quest for computer-based intelligence.

Can computers solve every problem?

To answer any question fairly, one has to understand it fully, one has to make sure that all the terms used in the question are defined with precision. Hence, if we want to know, for instance, whether or not all problems are solvable by computational means, we have to know the exact meaning of the terms "problem", "computer", and "computation". Only then we can attempt answering the question.

The discipline that can provide us with adequate definitions of the terms in question is Theoretical Computer Science (or TCS). It is one of the objectives of TCS to make the analysis of these definitions and to draw logical conclusions about computers from them.

The precise definitions of "problem", "computer", and "computation" as provided by TCS require a lot of background knowledge. Therefore in our discussion we shall mostly rely on approximate and, hence, imprecise definitions: a problem is a question that requires an answer, a solution is a correct answer to a problem, computing is a problem solving method using algorithms and computers (of today or any computer that will be built in the future).

TCS has matured enough to provide us with answers to at least some important questions about the power and limitations of computing. We know by now that unless we substantially modify our views on computers, the picture of computing that emerges is not as optimistic as we might want it to be.

Before I sketch this picture, let us simplify our task to investigate possible limits (or lack of) to computing by concentrating not on all problems but just problems concerning integer arithmetic, problems like

is there a value of X which makes the equation $X + 1 = 2$ true?

This problem is, of course, solvable and the answer (solution) is $X = 1$.

Question 1: How many problems concerning arithmetic are there?

Before we answer this question, let me explain why it is important. If the answer is: *the number of all problems concerning arithmetic is finite*, say one million, then we can just write one million computer programs to solve each problem individually and no question concerning arithmetic would be left unanswered.

Unfortunately, the number of problems concerning arithmetic is not finite. Please take a look at this never ending sequence of problems:

is there a value of X *which makes the equation* $X + 1 = 2$ *true?*
is there a value of X *which makes the equation* $X + 1 = 3$ *true?*
is there a value of X *which makes the equation* $X + 1 = 4$ *true?*
is there a value of X *which makes the equation* $X + 1 = 5$ *true?*
...
Clearly, every equation listed above can be solved in exactly the same way, following two simple steps:

1. subtract 1 from both sides of the equation (e.g. $X + 1 - 1 = 2 - 1$);
2. simplify both sides of the equation ($X + 1 - 1 = 2 - 1$ simplifies to $X = 1$).

Therefore a single computer program can handle all problems in this class. But how about other problems? In other words:

Question 2: Can a single computer program be written to solve all problems concerning arithmetic?

To answer this question, we have to review Alan Turing's work done in the 1930s and discussed in Lecture 5. Using his model of computing defined in terms of what we now call Turing Machines (and widely accepted as an adequate and most general) he demonstrated the existence of "unsolvable" problems – problems that cannot be solved algorithmically.

The negative answer to Question 2 may not be too disappointing if, for instance, the majority of problems are solvable and only a few are not.

Question 3: Which collection of problems concerning arithmetic is larger: the collection of solvable problems or unsolvable problems?

To answer this question we must know how to compare collections containing infinitely many items. A mathematical discipline that specifies how such a comparison can be done is called Set Theory. Using Set Theory and the notion of the Turing Machine one can conclude that, unfortunately, the collection of undecidable problems concerning arithmetic is much larger than the collection of decidable problems.

So far, the computational picture of our reality looks, from theoretical point of view, rather gloomy: the vast space of problems consists mostly of those that cannot be decided using computers. But wait, there is more bad news.

How much time, how much memory...

Since there is nothing that we can do about undecidable problems, let us concentrate on those that can. For these problems we can, informally speaking, write a computer program and execute it on a computer to get a solution. But there could be more trouble. What should be done when either:

(a) after 1 day, 10 days, a month, even a year, the computer is still working on a solution?

(b) no matter how much memory the computer is provided with, it will halt its computations and output the message "not enough memory to complete computation"?

It is therefore important to know how much "resources" the "best" computer would require to return a solution to a given problem. If the amount of memory required to solve a given problem equals, say, to the number of all atoms in the universe, then it would be irresponsible to even try to develop a computer program to solve this problem.

To deal with the issue of resources (e.g. time and memory), TCS has developed methods to estimate the "complexity" of a decidable problem. This notion captures the amount of time and memory required to execute theoretically the best program designed to solve a given problem. As a result, TCS classifies decidable problems as *feasible*, i.e. those that require "reasonable" resources, and non-feasible, i.e. those that cannot be solved due to resource requirements that we will not be able to satisfy.

Question 4: Which collection of problems concerning arithmetic is larger: the collection of feasible or non-feasible problems?

This question, too, has a disappointing answer: the collection of non-feasible problems is much larger than the collection of feasible problems.

Taking all the facts discussed so far into consideration, we can conclude that, from theoretical point of view, most problems are undecidable and of those which are decidable, most are too complex to be handled by computers. This is depicted in Fig. 2.

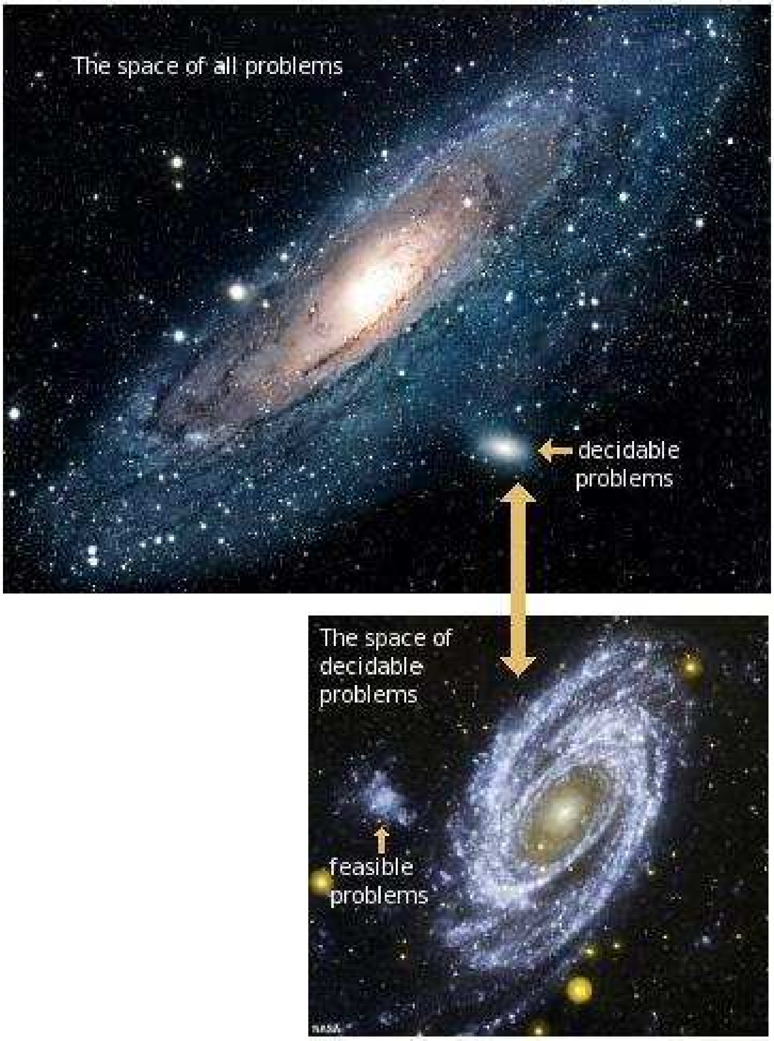

Fig. 2. The relationship between the classes of problems, decidable problems, and feasible problems.

Is all news bad?

Computers are the most complex and versatile problem solving tools ever invented. However, what follows from our discussion is that there is a limit to what we can do with them, what kind of problems can be practically solved.

The message is not all that bad. There are still infinitely many problems that can be efficiently solved with computers. For decidable but non-feasible problems, we can ask for approximate or partial solutions. Many cognitive tasks such as reasoning or planning are non-feasible. But, in spite of that, we do reason and do make plans in our every day activities. We overcome the ”computational complexity” of these tasks by incorporating a number of techniques (”tricks”) based on experience and intuition such as reasoning by analogy and default reasoning; we have learnt how to make ”good enough” decisions based on incomplete and, sometimes, inconsistent knowledge.

Of course it may turn out that our model of computation based on Turing machines is inadequate (although this is not likely); it is possible that somebody will propose new ways of looking at computation that will make some of the results discussed above invalid in general (while still valid for Turing machines, of course).

The quest for machine intelligence

The quest for mechanical intelligence has begun ever since humans embarked on constructing mechanical automata that exhibit some forms of human behaviour. In 1560, a person watching the Clockwork Prayer in action (the mechanical monk was discussed in Lectures 3 and 11) could naturally ask: can the monk understand me?

More than 60 years earlier, Leonardo da Vinci designed and constructed a robot knight. The mechanical warrior clad in medieval armour could stand, sit, raise its visor and independently move its arms. If a machine can imitate human-like motions, then, perhaps it would also be possible to advance the designs to create a machine capable of other functions: speech understanding, reasoning, planning, creative work.

Fig. 3. Leonardo da Vinci's artificial warrior reconstructed and displayed at Mensch - Erfinder - Genie exhibit, Berlin 2005. Photograph: Erik Möller.

Up until mid 20th century, mechanical intelligence was mostly the subject taken on by scammers and fiction writers. In the late 18th century, a Hungarian inventor Wolfgang von Kempelen designed and demonstrated a chess playing automaton called The Turk.

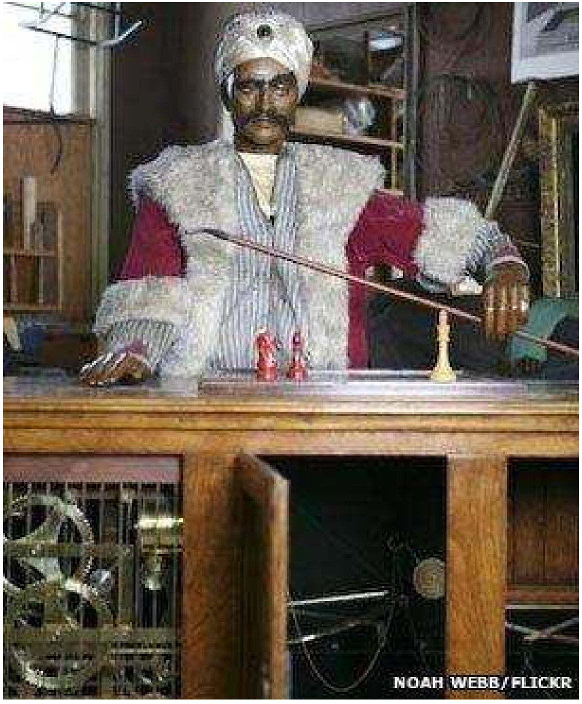

Fig. 3B. The Turk reconstructed. Source http://www.bbc.com/news/magazine-21876120

The machine, Kempelen claimed, could proficiently play chess and, indeed, until the machine's destruction in mid 19th century, The Turk managed to defeat many competent chess players including Napoleon Bonaparte and Benjamin Franklin. The success of the Turk seemed to indicate that man could design and built machines that exhibit at least some aspects of intelligence. Indeed, if a machine can understand board configurations and rules of the game and could use that knowledge to plan chess moves the to outplay a competent human chess player, then the machine has to be classified as intelligent. Unfortunately, those who lost to the Turk and those who truly believed in the invention did not know that the design was not a prelude to artificial intelligence but a hoax as the Turk was operated by a human player skillfully hidden inside the machine.

The next step in our search for the origins of artificial intelligence takes us to literature at the turn of the 20th century. While books written in this period did not offer any designs for "artificial brains", some of them would depicted characters on the boundary of real and artificially created life. In his 1883 children's book *The Adventures of Pinocchio*, Enrico Mazzanti created an unusual character. Pinocchio, a wooden puppet possesses its own intelligence (mostly focused on inventing lies). Eventually, Pinocchio is being transformed into a real boy (by a fairy, of course).

Fig. 4. Enrico Mazzanti's *The Adventures of Pinocchio.*
Source: http://www.linguaggioglobale.com/Pinocchio/menu_pinocchio.htm

Another literary work exploring the connection between real and artificial is L. Frank Baum's classic *The Wonderful Wizard of Oz* where we meet Tin Woodman, once a real man but, thanks to the wickedness of the Wicked Witch of the East, his body was entirely replaced with tin parts, leaving only emotions intact. Although his intelligence (mostly emotions) have human origins, his fusion into a mechanical automaton influenced others to go one step further and envision not only mechanical body but also intelligence.

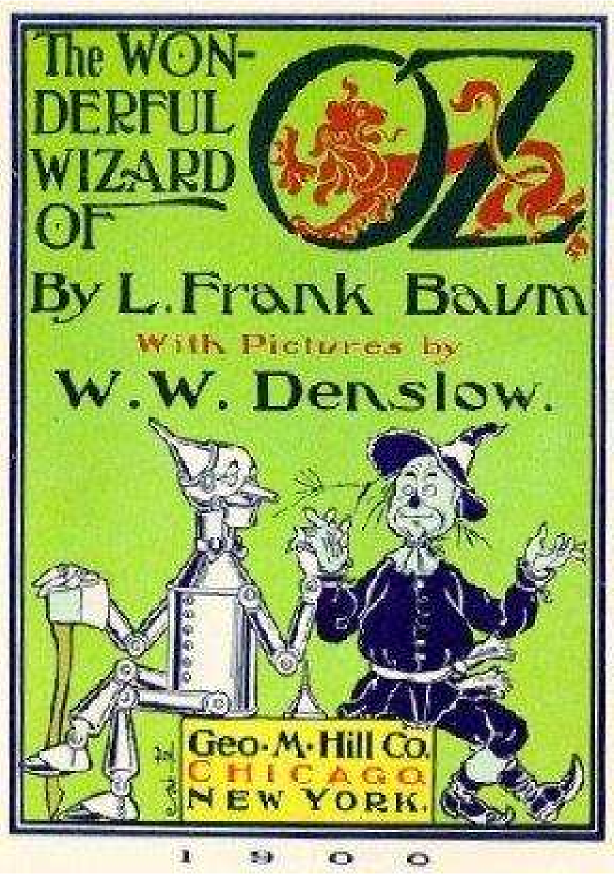

Fig. 5. The cover of L. Frank Baum's *The Wonderful Wizard of Oz*. Source: unknown.

In 1921, an unusual play was staged called R.U.R. (or "Rossum's Universal Robots") written by a Czech playwright Karel Čapek. Yes, there are robots in the play and, according to some scholars, it was that play that introduced the word "robot" (derived from the Polish "robota", meaning work) to the English language and to science fiction. In Čapek's play, robots are artificially manufactured people or androids equipped with artificial (i.e. non-human) intelligence. And yes, the robots rebel against humans and "eliminate" all of "us".

Fig. 6. RUR, Robot's rebellion. Source:
http://www.umich.edu/ engb415/literature/pontee/RUR/RURsmry.html

Robots invade Hollywood

Soon after the premiere of Čapek's play, robots "moved" to the cinema. In his 1927 futuristic movie *Metropolis*, German director Fritz Lang introduced a Machine-Man, a robot that was later transformed into a human.

Fig. 7. A 1927 *Metropolis* poster. Source: http://www.moviegoods.com/movie_product_static.asp?cmio=&sku=170580&master_movie_id=10337

In 1939, *The Wizard of Oz* was shown on a big screen. A year later Walt Disney studios showed its animated version of *The Adventures of Pinocchio* titled *Pinocchio.*

Since then, the fascination with computers, robots, and science fiction in general has resulted in myriad of films depicting artificial life and intelligence clashing with our own civilization. This trend will certainly continue as we advance our knowledge about intelligence and since the entertainment appetite for androids and intergalactic wars in movies and games shows no signs of being satisfied.

Fig. 7. A frame from the 1954 movie *The Bowery Boys Meet the Monsters.* Source: Allied Artists promotional photograph, York University Computer Museum.

HEARTBEEPS

Even robots can have maternal instincts, albeit electronic ones, as Aqua **(BERNADETTE PETERS)** leads Phil through the forest.

2136-6

Fig. 7. A frame from the 1981 movie *Heartbeeps.* "Even robots can have maternal instincts, albeit electronic ones, as Aqua leads Phil through the forest." Source: Universal City Studios Inc. promotional photograph, York University Computer Museum.

What is intelligence? what is artificial intelligence?

Early depictions and designs of robots and androids were modelled after human bodies and their (machine) intelligence was modelled after ours. As we have seen in the previous lectures, the early computers were presented by the industry and media as "giant brains" waiting for and obeying human instructions. Since intelligence is linked to specific brain functions, it was only natural to ask if some of these computer brains could eventually exhibit forms of intelligence, e.g. could deal with problems that we solve using our intelligence.

Fig. 8. Giant brains vintage cartoon. Source: unknown.

Clearly to answer this question one has to define the notion of intelligence in the first place, and that is not exactly a trivial task. Before we return to this issue, let us discuss Turing's view on machine intelligence.

We have already discussed Turing's contributions to the science of computing on many occasions; his impact on computer science was immense in terms of both the scope and depth. Early in this lecture we discussed the limits of computing and the pivotal role of Turing's research in establishing some groundbreaking results in Theoretical Computer Science.

But Turing is also known as a (if not "the") pioneer of Artificial Intelligence (AI) and his 1950 paper Computing Machinery and Intelligence published in *Mind* is considered a seminal work in AI (cf. [1]).

"I Propose to consider the question, 'Can machines think?'" begins Turing. Then he notes that thinking, like intelligence, cannot be satisfactory defined and, instead answering this question, he proposes an imitation game (now called the Turing test) and argues that a machine can be considered as a thinking device (or "intelligent") if it can pass the Turing test.

Turing's simulation game is played by three individuals: a man (A) a woman (B) and an interrogator (C). They are placed in separate rooms and can communicate in such a way that C cannot make distinction between A and B (say, they communicate via a communication network). C can ask anything and A and B can answer in any way they want – truthfully or not.

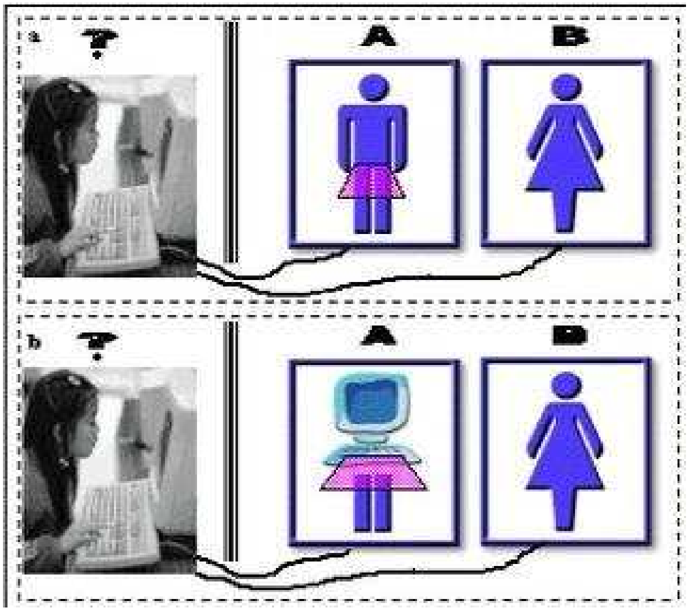

Fig. 10. Turing simulation game.

Source: http://physicaplus.org.il/zope/home/en/100846630031/1103975529_en

The role of the interrogator is to determine whether A is a man or a woman. Now, suppose that we replace A with a machine (or computer). If the interrogator C will identify incorrectly A as a woman as often as when A is a man rather than a woman, then the machine passes the test.

To sum up, the purpose of Turing simulation game is not to define intelligence but to provide a test for it.

The Turing test had a great impact on Artificial Intelligence as an area of research. Research programs were designed around passing the Turing test objective; a number of popular AI textbooks defined AI as a discipline focused on the study and design of computing artifacts that could contribute to the creation of a machine that can pass the test.

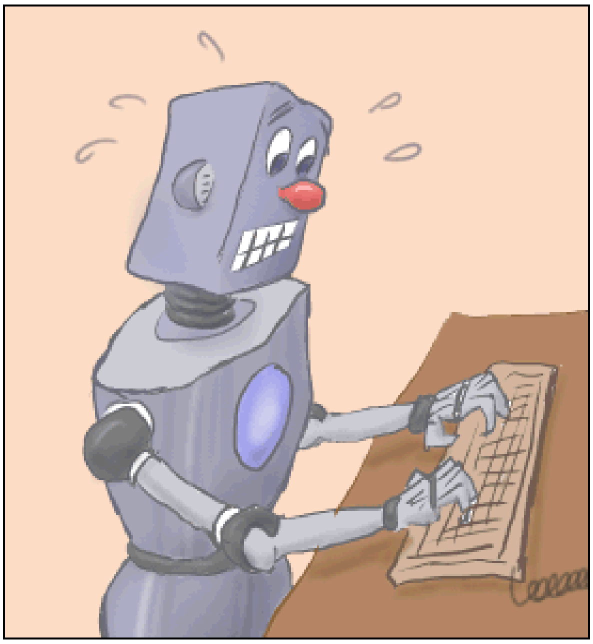

Fig. 10. A robot attempting Turing test. Source: www.zompist.com/crib.html

But as anybody involved in AI research can attest, the creation of machine intelligence Turing-style is a complex task. Such a machine has to be "good at", among other things, reasoning, planning, scheduling, learning, language processing, and even game playing. The machine has to posses skills such as arithmetic, searching, image processing and manipulation.

To implement tasks such as reasoning and planning or skills such as navigation and search, one has to make significant progress in several other research areas such as mathematics, physics, and linguistics, just to name a few. The issues of knowledge representation and manipulation, robotics, computer vision, communication and navigation have to be researched as well.

While early AI research was making some progress in most of these areas, the integration of the results into a single 'thinking machine' was not even on the AI horizon. Furthermore, some technology observers started to doubt whether anything of value could be achieved in spite of millions of dollars spent on AI research.

Eventually it became evident to the AI community that focusing on passing Turing test as the holy grail of AI was harmful, that such a focus was setting objectives that were neither beneficial nor plausible, delaying progress of the discipline rather than pushing it strongly forward. They saw little point in making 'thinking machines' in our own image a central focus of AI.

By the end of the 20th century, AI had shifted its research into discrete areas, focusing on specific aspects of intelligence, such as reasoning, knowledge representation, planning or language processing. With the new focus, the term "intelligence" gained a more precise meaning: instead of speaking about an intelligent machine, AI researchers have started to talk about machines being intelligent at solving a specific class of tasks, such as game playing or search.

With this new approach, intelligence could be ‘measured’ as a success rate at solving a problem. So, a machine is intelligent at chess playing if it wins a lot of games against skilled opponents. Another machine can be an intelligent reasoner, if, say, it can automatically prove some significant mathematical theorems, or if its reasoning skills can be successfully used for planning.

This new approach to defining the objectives of AI has resulted in an immense progress in AI research as well as in the development of a variety of practical applications some of which we shall discuss next.

References

1. A. Turing, Computing Machinery and Intelligence, *Mind*, October 1950.