Chapter 1 : Computer algebra - Wikipedia

*Artificial Intelligence: An MIT Perspective, Volume 2: Understanding Vision, Manipulation and Productivity Technology, Computer Design and Symbol Manipulation [Patrick Henry Winston, Richard H. Brown] on calendrierdelascience.com *FREE* shipping on qualifying offers.*

Overview Work in Artificial Intelligence AI has produced computer programs that can beat the world chess champion and defeat the best human players on the television quiz show Jeopardy. Our experience shows that playing chess or Jeopardy, and carrying on a conversation, are activities that require understanding and intelligence. Does computer prowess at challenging games and conversation then show that computers can understand and be intelligent? Will further development result in digital computers that fully match or even exceed human intelligence? By the late s some AI researchers claimed that computers already understood at least some natural language. Berkeley philosopher John Searle introduced a short and widely-discussed argument intended to show conclusively that it is impossible for digital computers to understand language or think. Searle argues that a good way to test a theory of mind, say a theory that holds that understanding can be created by doing such and such, is to imagine what it would be like to do what the theory says would create understanding. Searle summarized the Chinese Room argument concisely: Imagine a native English speaker who knows no Chinese locked in a room full of boxes of Chinese symbols a data base together with a book of instructions for manipulating the symbols the program. Imagine that people outside the room send in other Chinese symbols which, unknown to the person in the room, are questions in Chinese the input. And imagine that by following the instructions in the program the man in the room is able to pass out Chinese symbols which are correct answers to the questions the output. The program enables the person in the room to pass the Turing Test for understanding Chinese but he does not understand a word of Chinese. I demonstrated years ago with the so-called Chinese Room Argument that the implementation of the computer program is not by itself sufficient for consciousness or intentionality Searle Computation is defined purely formally or syntactically, whereas minds have actual mental or semantic contents, and we cannot get from syntactical to the semantic just by having the syntactical operations and nothing else. But such a specification necessarily leaves out the biologically specific powers of the brain to cause cognitive processes. A system, me, for example, would not acquire an understanding of Chinese just by going through the steps of a computer program that simulated the behavior of a Chinese speaker p. Thus Searle develops the broader implications of his argument. It aims to refute the functionalist approach to understanding minds, the approach that holds that mental states are defined by their causal roles, not by the stuff neurons, transistors that plays those roles. The argument counts especially against that form of functionalism known as the Computational Theory of Mind that treats minds as information processing systems. This interest has not subsided, and the range of connections with the argument has broadened. The first of these is an argument set out by the philosopher and mathematician Gottfried Leibniz â€" Moreover, it must be confessed that perception and that which depends upon it are inexplicable on mechanical grounds, that is to say, by means of figures and motions. And supposing there were a machine, so constructed as to think, feel, and have perception, it might be conceived as increased in size, while keeping the same proportions, so that one might go into it as into a mill. That being so, we should, on examining its interior, find only parts which work one upon another, and never anything by which to explain a perception. Thus it is in a simple substance, and not in a compound or in a machine, that perception must be sought for. He points out that these internal mechanical operations are just parts moving from point to point, hence there is nothing that is conscious or that can explain thinking, feeling or perceiving. For Leibniz physical states are not sufficient for, nor constitutive of, mental states. A paper machine is a kind of program, a series of simple steps like a computer program, but written in natural language e. The human operator of the paper chess-playing machine need not otherwise know how to play chess. All the operator does is follow the instructions for generating moves on the chess board. Turing was optimistic that computers themselves would soon be able to exhibit apparently intelligent behavior, answering questions posed in English and carrying on conversations. Turing proposed what is now known as the Turing Test: By the late s, as computers became faster and less expensive, some in the burgeoning AI community claimed that their programs could understand English sentences, using a database of background information. Berkeley colleague Hubert Dreyfus was an earlier critic of the claims made by AI researchers. Functionalists hold that mental states are defined by the causal role they play in a system just as a door stop is defined by what it does, not by what it is made out of. Critics of functionalism were quick to turn its proclaimed virtue of multiple realizability against it. In contrast with type-type identity theory, functionalism allowed beings with different physiology to have the same types of mental states as humansâ€”pains, for example. But it was pointed out that if aliens could realize the functional properties that constituted mental states, then, presumably so could systems even less like human brains. The computational form of functionalism is particularly vulnerable to this maneuver, since a wide variety of systems with simple components are computationally equivalent see e. Critics asked if it was really plausible that these inorganic systems could have mental states or feel pain. Daniel Dennett reports that in Lawrence Davis gave a colloquium at MIT in which he presented one such unorthodox implementation. Let a functionalist theory of pain whatever its details be instantiated by a system the subassemblies of which are not such things as C-fibers and reticular systems but telephone lines and offices staffed by people. Perhaps it is a giant robot controlled by an army of human beings that inhabit it. That is, real pain, as real as our own, would exist in virtue of the perhaps disinterested and businesslike activities of these bureaucratic teams, executing their proper functions. No phone message need be exchanged; all that is required is the pattern of calling. The phone calls play the same functional role as neurons causing one another to fire. Block was primarily interested in qualia, and in particular, whether it is plausible to hold that the population of China might collectively be in pain, while no individual member of the population experienced any pain, but the thought experiment applies to any mental states and operations, including understanding language. In this article, Searle sets out the argument, and then replies to the half-dozen main objections that had been raised during his earlier presentations at various university campuses see next section. In the decades following its publication, the Chinese Room argument was the subject of very many discussions. In January , the popular periodical Scientific American took the debate to a general scientific audience. Soon thereafter Searle had a published exchange about the Chinese Room with another leading philosopher, Jerry Fodor in Rosenthal ed. The human produces the appearance of understanding Chinese by following the symbol manipulating instructions, but does not thereby come to understand Chinese. Since a computer just does what the human doesâ€”manipulate symbols on the basis of their syntax aloneâ€—no computer, merely by following a program, comes to genuinely understand Chinese. Strong AI is the view that suitably programmed computers or the programs themselves can understand natural language and actually have other mental capabilities similar to the humans whose behavior they mimic. According to Strong AI, these computers really play chess intelligently, make clever moves, or understand language. But weak AI makes no claim that computers actually understand or are intelligent. The Chinese Room argument is not directed at weak AI, nor does it purport to show that no machine can thinkâ€”Searle says that brains are machines, and brains think. The argument is directed at the view that formal computations on symbols can produce thought. We might summarize the narrow argument as a reductio ad absurdum against Strong AI as follows. A computing system is any system, human or otherwise, that can run a program. If Strong AI is true, then there is a program for Chinese such that if any computing system runs that program, that system thereby comes to understand Chinese. I could run a program for Chinese without thereby coming to understand Chinese. Therefore Strong AI is false. The second premise is supported by the Chinese Room thought experiment. The conclusion of this narrow argument is that running a program cannot endow the system with language understanding. That and related issues are discussed in the section The Larger Philosophical Issues. Replies to the Chinese Room Argument Criticisms of the narrow Chinese Room argument against Strong AI have often followed three main lines, which can be distinguished by how much they concede: These critics object to the inference from the claim that the man in the room does not understand Chinese to the conclusion that no understanding has been created. There might be understanding by a larger, or different, entity. But these critics hold that a variation on the computer system could understand. For example, critics have argued that our intuitions in such cases are unreliable. Sprevak object to the assumption that any system e. Searle in the room can run any computer program. The objection is that we should be willing to attribute understanding in the Chinese Room on the basis of the overt behavior, just as we do with other humans and some animals , and as we would do with extra-terrestrial Aliens or burning bushes or angels that spoke our language. As a result, these early responses have received the most attention in subsequent discussion. The Systems Reply, which Searle says was originally associated with Yale, concedes that the man in the room does not understand Chinese. But, the reply continues, the man is but a part, a central processing unit CPU , in a larger system. The larger system includes the huge database, the memory scratchpads containing intermediate states, and the instructionsâ€”the complete system that is required for answering the Chinese questions. So the Sytems Reply is that while the man running the program does not understand Chinese, the system as a whole does. Rey says the person in the room is just the CPU of the system. Kurzweil says that the human being is just an implementer and of no significance presumably meaning that the properties of the implementer are not necessarily those of the system. Margaret Boden raises levels considerations. Boden points out that the room operator is a conscious agent, while the CPU in a computer is notâ€”the Chinese Room scenario asks us to take the perspective of the implementer, and not surprisingly fails to see the larger picture. He could then leave the room and wander outdoors, perhaps even conversing in Chinese. The man would now be the entire system, yet he still would not understand Chinese. For example, he would not know the meaning of the Chinese word for hamburger. He still cannot get semantics from syntax. See below the section on Syntax and Semantics. Copeland then turns to consider the Chinese Gym, and again appears to endorse the Systems Reply: But there is no entailment from this to the claim that the simulation as a whole does not come to understand Chinese. Copeland denies that connectionism implies that a room of people can simulate the brain. According to Haugeland, his failure to understand Chinese is irrelevant: The larger system implemented would understandâ€”there is a level-of-description fallacy. Shaffer examines modal aspects of the logic of the CRA and argues that familiar versions of the System Reply are question-begging. But, Shaffer claims, a modalized version of the System Reply succeeds because there are possible worlds in which understanding is an emergent property of complex syntax manipulation. Nute is a reply to Shaffer. Is the mind a computer program? Or, more specifically, if a computer program simulates or imitates activities of ours that seem to require understanding such as communicating in language , can the program itself be said to understand in so doing? However the Virtual Mind reply holds that what is important is whether understanding is created, not whether the Room operator is the agent that understands.

DOWNLOAD PDF V. 2. UNDERSTANDING VISION, MANIPULATION, COMPUTER DESIGN, SYMBOL MANIPULATION.

Chapter 2 : The Chinese Room Argument (Stanford Encyclopedia of Philosophy)

Review of "Artificial Intelligence: An MIT Perspective, Volume 2: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation by Patrick Henry Winston and Richard Henry Brown"; The MIT Press, Cambridge, Massachusetts, Second printing, , ISBN

Terminology[edit] Some authors distinguish computer algebra from symbolic computation using the latter name to refer to kinds of symbolic computation other than the computation with mathematical formulas. Some authors use symbolic computation for the computer science aspect of the subject and "computer algebra" for the mathematical aspect. Typically, it is called calcul formel in French, which means "formal computation". This name reflects the ties this field has with formal methods. Symbolic computation has also been referred to, in the past, as symbolic manipulation, algebraic manipulation, symbolic processing, symbolic mathematics, or symbolic algebra, but these terms, which also refer to non-computational manipulation, are no longer used in reference to computer algebra. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. November Data representation[edit] As numerical software is highly efficient for approximate numerical computation , it is common, in computer algebra, to emphasize exact computation with exactly represented data. Such an exact representation implies that, even when the size of the output is small, the intermediate data generated during a computation may grow in an unpredictable way. This behavior is called expression swell. To obviate this problem, various methods are used in the representation of the data, as well as in the algorithms that manipulate them. Numbers[edit] The usual numbers systems used in numerical computation are either the floating point numbers and the integers of a fixed bounded size. None is convenient for computer algebra, because of the expression swell. Therefore, the basic numbers used in computer algebra are the integers of the mathematicians, commonly represented by an unbounded signed sequence of digits in some base of numeration , usually the largest base allowed by the machine word. These integers allow to define the rational numbers , which are irreducible fractions of two integers. Programming an efficient implementation of the arithmetic operations is a hard task. Therefore, most free computer algebra systems and some commercial ones, like Maple software , use the GMP library , which is thus a de facto standard. Expressions[edit] Except for numbers and variables , every mathematical expression may be viewed as the symbol of an operator followed by a sequence of operands. In computer algebra software, the expressions are usually represented in this way. This representation is very flexible, and many things, that seem not to be mathematical expressions at first glance, may be represented and manipulated as such. Conversely, any mathematical expression may be viewed as a program. Executing this program consists in evaluating the expression for given values of a and b; if they do not have any valueâ€”that is they are indeterminatesâ€”, the result of the evaluation is simply its input. This process of delayed evaluation is fundamental in computer algebra. As the size of the operands of an expression is unpredictable and may change during a working session, the sequence of the operands is usually represented as a sequence of either pointers like in Macsyma or entries in a hash table like in Maple. The raw application of the basic rules of differentiation with respect to x on the expression a.

Chapter 3 : - NLM Catalog Result

Get this from a library! Artificial Intelligence: an MIT perspective / Vol. 2, Understanding vision, manipulation, computer design, symbol manipulation.