題名	人工知能の論理と哲学
Title	The Logic and Philosophy of Artificial Intelligence
著者名	ジョン・マッカーシー
Author(s)	John McCarthy
言語 Language	日本語・英語　Japanese, English
書名	稲盛財団：京都賞と助成金
Book title	The Inamori Foundation: Kyoto Prizes & Inamori Grants
受賞回	4
受賞年度	1988
出版者	財団法人 稲盛財団
Publisher	The Inamori Foundation
発行日 Issue Date	8/20/1992
開始ページ Start page	94
終了ページ End page	127
ISBN	978-4-900663-04-2

人工知能の論理と哲学

ジョン・マッカーシー

人工知能のゴールは、知能を必要とする問題を解き、目標を達成することにおいて、人間よりも有能な機械であります。有用な結果が得られてはいるのですが、最終的なゴールは、まだ大きな概念的進歩を要求しており、その達成は、はるかに遠いものと考えられます。

ゴールに攻め込む道は三つあり、第一の道は人間の神経システムをまねること、第二の道は人間の知能の心理学を研究すること、そして、第三の道は人々がその中で目標を達成しようとしているような常識の世界を理解して、知能的な計算機プログラムを作ることであります。この第三の道が計算機科学としての接近方法であって、これが現在最も成功している方法であり、これについてこの講演の中で議論したいと思います。

人工知能研究者の間でも、機械が世界に関して知っていることを表現するために数理論理の言葉を使うことをどの程度重く見るべきかについて意見が異なっています。手短に言えば、事実については互いに独立に、またそれらが使われることとも独立に学習することができるという利点を持っています。反対の意見としては、人間の推論はそんなに論理的でないと思われる点についてであります。現在のエキスパートシステム技術では、その知識の大部分を表現するために論理的な言語が使われていますが、それ以外に多くのことが計算機プログラムの中に埋め込まれており、また論理式の並べ方についても多くのことがなされています。私はこの講演の中で論理学的接近の方法を強調したいと考えております。そしてこの機会にふさわしいように私自身の研究も強調したいと思います。

1948年9月に、行動における大脳のメカニズムに関するヒクソン・シンポジウム（Jeffress 1951）のいくつかの分科会に出席したときから、人工知能に興味を持つようになりました。それは、カリフォルニア工科大学で開催されたものであり、また、私が数学科の大学院で勉強を始めたばかりのときでありました。このシンポジウムには、数学者ジョン・フォン・ノイマン（プログラム貯蔵式電子計算機の発明者の一人）、ワレン・マッカロッチ（仮想的神経網が計算のためにどのように使われるかについて話していた）と著名な心理学者、神経生理学者が参加していました。フォン・ノイマンの講演の題は、〝オートマトンの一般的論理学的理論〟でありました。人間の脳に関してわかっていることと、電子計算機の新しい考え方との比較について、たいへん興味をひかれました。この時期には、プログラム貯蔵式電子計算機は建設中でありまして、1号機の完成はその翌年になります。

最近、このシンポジウムの議事録を読みなおしてみましたが、私の記憶は誤ってお

THE LOGIC AND PHILOSOPHY OF ARTIFICIAL INTELLIGENCE

John McCarthy

The goal of artificial intelligence (A.I.) is machines more capable than humans at solving problems and achieving goals requiring intelligence. There has been some useful success, but the ultimate goal still requires major conceptual advances and is probably far off.

There are three ways of attacking the goal. The first is to imitate the human nervous system. The second is to study the psychology of human intelligence. The third is to understand the common sense world in which people achieve their goals and develop intelligent computer programs. This last one is the computer science approach. It is the one that has had the most success so far, and it is the one I will discuss in this lecture.

Among A.I. researchers, opinions differ about how much to emphasize mathematical logical languages for expressing what the machine knows about the world. Briefly, the advantage is that facts can be learned independently of one another and of the use to which they will be put. The argument against it is that much human reasoning doesn't seem very logical. The current expert system technology uses logical languages for expressing much of their knowledge, but a lot is built into computer programs and also into the arrangement of the logical sentences. I will emphasize the logical approach in this lecture. As perhaps befits the occasion, I will emphasize my own work.

I became interested in artificial intelligence in September, 1948 when I attended some sessions of the Hixon Symposium on Cerebral Mechanisms in Behavior (Jeffress 1951) held at the California Institute of Technology where I was starting graduate work in mathematics. The symposium included the mathematician John Von Neumann (one of the inventors of the stored program computer), Warren McCulloch (who had shown how networks of hypothetical neurons could be used for computing) and many famous psychologists and neurophysiologists. Von Neumann's paper was entitled "The General and Logical Theory of Automata," and there was much interest in comparing what was known about the human brain with the new idea of an electronic computer. At that time, stored program computers were under construction, but the first wasn't finished until the following year.

When I recently re-examined the proceedings of the symposium, I discovered that my memory was incorrect and nothing had been said about trying to make intelligent computers. All the famous participants were fully committed to their existing research programs in biology, psychology and mathematics. Von Neumann discussed two ideas that he later developed more fully—how to make

りまして、知能的計算機を作る試みについては、何も触れられておりませんでした。すべての著名な出席者は、生物学、心理学、および、数学の在来型の研究についてのみ語っていました。フォン・ノイマンは、後にさらに十分発展した二つのアイデア、信頼性の低い部品からいかにして信頼性の高い計算機を作るかという話と、自己増殖機械をどのようにして作るかについて話しています。彼は数理論理学について重要な研究を行った人ですが、このときも、そしてその後も、世界についての事実を論理式で表現することについては何も議論しておりません。彼は計算機の開発には積極的に参加していましたが、計算機が知的に振る舞うためのプログラムの作り方については議論していません。いずれにいたしましても、この会議は私の学問的野心を人工知能へ向かわしめたものでありました。

私は1948年に人工知能について考え始めたわけですが、計算機のプログラムの作り方や、事実を論理式で表現することを考えたわけではありません。私は論理学についてもまた計算機についてもあまり知りませんでした。そのかわり、私は、後に袋小路であったと思うようになった、二つの考え方を追っていました。一つは、オートマトン理論と情報理論とを結びつけようとするもので、この二つの理論は当時新しくて流行的なものでありました。脳は環境に結びつけられた有限オートマトンであり、環境もまた有限オートマトンであると考えていました。環境が何であるかについて脳がはっきりしていないという事実を表現するために、私は環境オートマトンのアンサンブル（確率のついた集合）を考えていました。情報理論はこのアンサンブルに適用され、脳が最初に環境に結びつけられた時である時刻0の時にエントロピーを定義することが許され、その後システムが作動してしばらく経ってから、脳の状態はアンサンブルからどんな環境が選ばれているかに部分的に依存していることになります。これらの時刻のエントロピーの差が、どれだけ脳が環境について学んだかの物差しになります。

その後私はこれがまずい考え方であると信ずるようになり、発表することを試みませんでした。困難な点は、世界についての固有の事実を環境のアンサンブルを使ってどのように表現してよいのかわからなかったことであります。後に得られた用語を先回りして使うと、このような表現方法は認識論的に不適切であったということであります。それは実際に得られる情報を表現できていませんでした。

私のもう一つの良くないアイデアは、より数学的なものでありました。問題が何であるかということ、すなわち知的な機械に解いてもらいたくなるような問題は何かということであります。私は〝はっきり定義された問題〟は、提起された解をテストするための方法によって与えられることを提案しました。それは良い考え方であって、

reliable machines out of unreliable components, and how to make self-reproducing machines. Although he had done important work in mathematical logic, he didn't discuss, then or later, representing facts about the world in logic. Although he was actively involved in developing computers, he didn't discuss programming them to behave intelligently. Nevertheless, the conference oriented my own scientific ambitions toward artificial intelligence.

While I started thinking about A.I. in 1948, I didn't think about programming computers or about representing facts in logic. I didn't know much about either logic or computers. Instead I successively pursued two ideas that I later came to regard as blind alleys. The first combined automata theory and information theory, both newly fashionable. It considered a brain as a finite automaton connected to an environment also considered as an automaton. To represent the fact that the brain is uncertain about what the environment is like, I considered an ensemble (i.e. a set with probabilities) of environment automata. Information theory applied to this ensemble permitted defining an entropy at time 0 when the brain was first attached to the environment and later when the system had run for a while and the state of the brain was partially dependent on which environment from the ensemble had been chosen. The difference of these entropies measured how much the brain had learned about the environment.

I came to believe that this was a bad idea and never tried to publish it. The trouble was that I couldn't see how to represent specific facts about our own world in terms of an ensemble of environments. To anticipate later terminology, the representation was epistemologically inadequate. It couldn't represent the information actually available.

My next bad idea was more mathematical. What is a problem, i.e. one that we might want an intelligent machine to solve? I proposed that a "well-defined problem" is given by a method for testing proposed solutions. That's a good idea and has survived in one form or another, but maybe it's obvious. The method of testing proposed that solutions have to be well defined, so let it be given by a Turing machine — the abstract form of computer introduced by Alan Turing in 1936. Therefore, problem solving may be regarded as the problem of inverting functions defined by Turing machines. I wrote a paper (McCarthy 1956) on this while working on a summer job with Claude Shannon in 1952, and it was included in our Automata Studies volume published as (Shannon and McCarthy 1956).The trouble is that this is also epistemologically inadequate.

That stored program computers were the right vehicle for developing artifi-

一つの形、または違った形で存在し続けたのですが、それは明白すぎることであったかもしれません。提起された解をテストする方法は明確に定義されねばなりませんから、チューリング機械で与えることにしました。チューリング機械は1936年に計算機の抽象的な形としてアラン・チューリングによって導入されたものであります。従って、問題解決はチューリング機械によって定義される関数の逆関数を求める問題とみなすことができます。1952年にクロード・シャノンの下で夏休みに働いている間にこのことについて論文（マッカーシー　1956）を書きました。そしてオートマタ・スタディーズ（シャノンとマッカーシー　1956）に入れていただくことにより発表されました。困難点は、やはり認識論的に不適切であることでした。

プログラム貯蔵式計算機が、人工知能の展開に適切な手段であるか否かは、私にとって、まだそれほど明白なことではありませんでしたし、また、オートマタ・スタディーズのどの著者にとってもそうでした。

このことを最初に明確に理解した人は、アラン・チューリングでした。彼の論文〝計算機と知能〟（1950）は哲学誌に発表され、エドワード・R・ニューマンによって〝数学の世界〟（1956）に再度載せられるまで知られておりませんでした。私も独自に同じアイデアに到達しましたが、アレン・ニューエル、ハーバート・サイモンもそうでした。おそらくニューエルが計算機が知的な振る舞いをするようにプログラムを作成することを、生涯の仕事とした最初の人でありましょう。チューリングは1954年、計算機が人工知能に貢献しうるようになる前に死去しました。

私の計算機との出会いは、1955年、IBMのナサニエル・ロケスターが、パキプシーにある彼の先駆的な情報研究部に夏の間雇ってくれたときのことでした。私はIBM702のプログラムを作ることを学び、人工知能に対する私の考え方を知能的な計算プログラムを作ることに向けてくれました。私は当時ダートマス大学数学科の助教授でありましたが、1956年に人工知能に興味を持つ人々をダートマスに招待する夏の研究プロジェクトを行うのに、十分なだけ人工知能には興味があるということが、その夏わかったような次第でした。そんなにたくさんの人々を招待できたわけではありませんでしたが、興味を持ってほしいと望んでいた人々を招待することができました。クロード・シャノン、マービン・ミンスキー、オリバー・セルフリッヂ、ナサニエル・ロケスターがロックフェラー財団への提案に協力してくれました。そして、旅費、生活費等にあてる資金7,500ドルが与えられました。

このときに我々のテーマを決めねばなりませんでしたが、私は〝人工知能〟を選びました。このような大胆な題名を選んだのは、参加者に望ましい影響を与えたいと思っ

cial intelligence was still not apparent to me or to any of the others who contributed to *Automata Studies*.

The person to whom it was first apparent was Alan Turing. His (1950) paper "Computing Machinery and Intelligence" was in a philosophy journal and didn't become well known until it was reprinted by Edward R. Newman in his *World of Mathematics* in 1956. I had to come to the idea independently and so did Allen Newell and Herbert Simon. Newell was probably the first person to make programming computers to behave intelligently a career. Turing died in 1954, before computers were good enough to do much in the way of A.I..

My introduction to computers came in 1955, when Nathaniel Rochester of IBM hired me for the summer in his pioneering Information Research Department in Poughkeepsie. I learned to program the IBM 702, and re-oriented my thinking about A.I. to making intelligent computer programs. I was an assistant professor of mathematics at Dartmouth College then, and it occurred to me that summer that there was enough interest in A.I. to warrant a summer research project at Dartmouth in 1956 that would invite everyone we knew who might be interested. That wasn't too many, and we were able to invite a number of people that we only hoped would be interested. Claude Shannon, Marvin Minsky, Oliver Selfridge and Nathaniel Rochester joined me in making a proposal to the Rockefeller Foundation, and they awarded us $7,500 to pay for some travel and living expenses.

We had to decide what to call the subject, and I chose "artificial intelligence." The reason for choosing such a bold title was the desired effect on participants. When Shannon and I collected papers for *Automata Studies*, it seemed to me that we got too many papers treating the mathematical properties of automata that were only peripherally relevant to making intelligent automata. The name "artificial intelligence" nailed the flag to the mast. Many people have criticized the name, but I think it was and still is important to have a name that makes people compare their present projects with the ambitious long term goal.

To me the most interesting work presented at Dartmouth was the Newell-Simon "Logic Theory Machine." Their program simulated the processes that naive student subjects went through in solving problems in elementary logic. In the course of this, they also developed the IPL list processing language and introduced recursive subroutines. Other interesting projects were Arthur Samuel's program for the game of checkers and Alex Bernstein's projected program for chess, both IBM projects. Marvin Minsky proposed a theorem prover for plane

たからです。シャノンと一緒に、オートマタ・スタディーズのために論文を集めたのですが、そのとき私は知能的オートマタを作るためには周辺的な部分にしか関係していないような、オートマタの数学的性質を扱っている論文が多すぎるように思いました。〝人工知能〟という名前は、帆柱に掲揚された旗でありました。多くの人々はこの名前に批判的でしたが、人々に彼らの現在のプロジェクトと野心的な長期的ゴールとを比べさせるような名前を持つことが、過去においても、また現在でも重要なことであると私は考えております。

ダートマスにおいて発表された論文のうち私にとって最も興味深く思えたのは、ニューエルとサイモンの〝数理論理機械〟(Logic Theory Machine) でありました。彼らのプログラムは、普通の学生が初等論理学の問題を解いていくプロセスをまねるものでした。このプログラムの作成過程でIPLリスト処理言語を開発し、また再帰的サブルーチンを初めて導入したのです。この他の興味あるプロジェクトとしては、アーサー・サムエルのチェッカーのゲームのためのプログラム、アレックス・ベルンシュタインのチェスのためのプログラムがありました。これらはIBMのプロジェクトでした。マービン・ミンスキーは平面幾何の定理証明機を提案し、計算機に〝図形〟を記憶させておいて、どの命題が証明を試みるのに十分価するかを決めることに使うようにすべきであると言っていました。私自身は、事実を証明するために論理を使うことを考え始めていましたが、ダートマスにおける会合で発表するに十分なものは持っていませんでした。

この会合の後、ロチェスターは幾何用定理証明機に関するミンスキーのアイデアのプログラムを作ることを決め、ヘルベルト・ゲレルンターをプログラム作成のため、また私をコンサルタントとして雇いました。私はリスト処理言語に関するニューエル・サイモンのアイデアに深い印象を受けておりましたが、しかし、彼らの使った特定の形式的枠組みが私に訴えるものは何もありませんでした。IBMのプロジェクトであるFORTRANのほうがより魅力的な形を持っていました。そこで私は、ゲレルンターに対して、幾何のためのリスト処理をFORTRANで作ることを提案し、いくつかの基本関数——LISPのcarとcdr——を提案しました。ゲレルンターとカール・ゲルベリッヒがそれを作るとともに、さらにいくつかの関数をつけ加えました。特にconsを関数として作り、従って合成することができるようにしました（このことは、当時必ずしも明白なことではなかったと、あなた方に納得してもらえるような知的情況を説明するスペースを、私は持ち合わせておりません)。

1957年秋にMITに行くまでは、私はフルタイムの計算機科学者ではありませんでし

geometry that used a "diagram" in the memory of the computer to decide which sentences were plausible enough to try to prove. I had started thinking about the use of logic to represent facts, but I didn't have much to present at the Dartmouth meeting.

After the meeting, Rochester decided to implement Minsky's ideas of the geometry theorem prover and hired Herbert Gelernter to do it, with me as consultant. I had been impressed by the Newell-Simon idea of a list processing language, but their particular formalism didn't appeal to me at all; IBM's projected Fortran had a much more attractive style. Therefore, I proposed to Gelernter that he do his list processing for geometry in Fortran and proposed some basic functions — the *car* and *cdr* of LISP. Gelernter and Carl Gerberich did it, and added something more — making *cons* a function so it could be composed. (I don't have the space to describe the intellectual situation well enough to convince you that this wasn't the one obvious thing to do.)

I didn't become a full-time computer scientist until I went to MIT in Fall 1957. By then I was convinced that logic was the key formalism for artificial intelligence. While Newell and Simon (and several others soon after) had already written programs to do logic, none of them were interested in using logic except as a subject domain. My conversion to logic was expressed in my 1958 paper "Programs with Common Sense" (McCarthy 1959).

In 1958 I spent another summer with Rochester's IBM group, where Gelernter and Gerberich were finishing their plane geometry program. I became convinced that Fortran didn't permit the expression of recursive list processing functions and started work on formulating LISP. When I returned to MIT in fall 1958, Minsky and I were given a big work room, a key punch, a secretary, two programmers and six mathematics graduate students. This was a very decisive and prompt action by Jerry Wiesner, considering that we had asked for these things the preceding spring in order to form an artificial intelligence project and hadn't even prepared a written proposal. Fortunately, the Research Laboratory of Electronics at MIT had just been given an open-ended Joint Services Contract by the U.S. Armed Forces and hadn't yet committed all the resources. I think that such flexibility is one of the reasons the U.S. started in A.I. ahead of other countries. The Newell-Simon work was also possible because of the flexible support the U.S. Air Force provided the Rand Corporation.

In September 1958 we started on LISP, and the first interpreter was working in early 1959. The first idea was just a Fortran-like language with list structures

た。そのころまでには、私は論理が人工知能に対する鍵となる一つの形式であると確信するようになっていました。ニューエルとサイモン（そして間もなく他の数人の人々）は、論理を実行するプログラムをすでに作り上げていましたけれど、誰一人その主題の領域として考える以外は、論理を使うことに興味を持ってはいませんでした。私の論理への転向は、1958年の論文〝常識を持つプログラム〟（マッカーシー　1959）に述べてあります。

1958年夏を再びロチェスターのIBMグループと過ごしましたが、そこで、ゲレルンターとゲルベリッヒは、平面幾何のプログラムを完成しようとしていました。私は、FORTRANが再帰的リスト処理関数を良い方法で表現していないと確信するようになり、LISPを定式化する研究を始めました。1958年秋にMITに帰ったとき、ミンスキーと私とには、大きな研究室と、キーパンチ機１台と一人の秘書、二人のプログラマーと６人の数学の大学院生とが与えられました。私たちがその前の春に人工知能のプロジェクトを作るためにお願いしてあったことを考えて、文書による要望はまだ出してもいないのに、ジェリー・ウィズナーが決定的かつ速やかな行動をとったのでありました。幸いにして、MITの電子工学研究所が米国空軍と制限なしの共同研究契約を結んだばかりのときであり、しかも、まだ財源の使用計画がすべて決まっているわけではないというときでありました。このような柔軟性が、米国が他の国々にさきがけて人工知能の研究をスタートさせた理由であると私は考えています。ニューエルとサイモンの研究を可能にしたのも、米国空軍がランド・コーポレーションに与えた柔軟性のある援助によるものでありました。

1958年９月に我々はLISPの研究を始め、そして、1959年初頭には一番最初のインタプリターが稼働していました。最初の考え方はFORTRAN類似の言語でリスト構造をデータとして持つものでありました。しかしながら、大学１年生に対して普通教えられるような代数式の微分のプロセスの論理構造を注目すると、関数を帰納的に使えること、条件式が与えられること、ガーベーヂ・コレクション（ごみ集め）を使うことにより、直接な消去を避けることができること等が可能な場合には、代数式の微分を行う関数は、一つの簡単な式に表させるということが判明しました。この前年FORTRANで書いたチェスのプログラムに関連して私は条件式を発明しました。ラムダ式は、微分の例から示唆を得たもう一つの道具でした。数学者に、普通計算機は、チューリング機械よりもLISPを使うほうがよりコンパクトに記述できることを示すために、私は関数EVALを書きました。我々のプログラマーの一人であったスティーブ・ラッセルは、それを速やかにLISP用のインタプリターに変換しました。「マッカーシ

as data. However, a look at the logical structure of the process of differentiating algebraic expressions, as normally described to college freshmen, showed that that function could be described by a simple formula provided it could be used recursively, provided conditional expressions were allowed, and provided explicit erasure could be avoided by using garbage collection. I had invented conditional expressions the previous year in connection with a chess program written in Fortran. Lambda expressions were another tool whose use was suggested by the differentiation example. In order to show mathematicians that a universal computer could be described much more compactly in LISP than with Turing machines, I wrote the function EVAL. Steve Russell, one of our programmers, who promptly turned it into an interpreter for LISP (McCarthy 1977a), describes this history.

Early in 1979 James Slagle started his work on symbolic integration under Minsky's supervision. About this time, the promising IBM A.I. work was shut down, and IBM didn't really resume A.I. work until 1983 and still plays a minor role. The reason for shutting it down was apparently a combination of uninformed scientific criticism and public relations. IBM wanted computers to have the image of mere data processors—nothing revolutionary was wanted.

The work in A.I. also led to my first proposals for time-sharing computer systems. Early ideas about computers emphasized running a program for a long time. The programs were imagined to be numerical. The process of developing the program was considered auxiliary and so was any interaction with it. It seemed to me that artificial intelligence required a quite different approach. Someone doing A.I. research might spend almost all of his time developing the program and interacting with it. He needed to sit at a terminal in his own office and interact at his own pace and convenience, rather than sign up for half an hour at 2 a.m. or submit decks of cards. Again it would require a lot of explanation today to convince you that the idea wasn't obvious. Not only wasn't it obvious, but it encountered considerable resistance. For example, the IBM developers of the 360 knew about the idea but didn't believe it. Fortunately, the Digital Equipment Corporation developers of the PDP-6 computer (somewhat later) had been MIT students and did time-sharing from the start.

Actually there were two approaches to providing on-line computation. One was time-sharing, in which a large computer switched its time among users, and the other was the personal computer, pioneered at MIT Lincoln Laboratories by Wes Clark, who created the TX-0 and TX-2 computers, both extremely expensive

ー　1977a」にはこの歴史が述べてあります。

1979年初期に、ジェームス・スレーグルは、ミンスキーの指導のもとで、記号積分についての研究を始めました。このころに成果の期待されていたIBMの人工知能の研究が中止になり、その後IBMは1983年までその研究を再開しませんでしたし、まだ小さな役割しか果たしておりません。その中止の理由は、明らかに無知な科学的批判とP.R.との組み合わせでありました。IBMは、計算機が単なるデータ処理機のイメージを持つべきであることを望んでおり、革命的なことは何も望んでおりませんでした。

AIに関する研究は、また時分割方式計算機システムについての最初の提案へ私を導きました。以前の計算機に関する考え方は、プログラムを長時間走らせることに重点をおいていました。そのプログラムは数字で表されたものと考えられていました。プログラムの開発のプロセスは副次的なものであると考えられており、従ってこのプロセスでの計算機との対話も同様にみなされておりました。私には、人工知能はまったく異なる接近方法を要求しているように思えました。人工知能の研究を行っている人はその時間の大部分をプログラムの開発と、計算機との対話に費やしていたと思います。研究者には自分の研究室でターミナルの前に座って、自分のペースで自分の都合のよいときに対話することが、午前2時に30分使うことを申し込んだり、カード・デックを納めることよりも、必要なのでした。ここでもこのアイデアが明白なことではないということを信じていただくには、多くの説明を必要とするでありましょう。明白でなかっただけでなく、非常に多くの批判に出会いました。例としては、IBM360の開発者たちは、このアイデアを知ってはいましたが、それを信用しようとしませんでした。幸いにして、デジタル・イクイプメント株式会社のPDP-6という（少し後の）計算機の開発者たちは、MITの卒業生であり、最初から時分割方式を採用しました。

実際にはオン・ライン計算機を与えるには、二つの方式がありました。一つは時分割方式であって、大型の計算機をその計算時間を分割して多くの使用者にスイッチしながら与えるものであり、もう一つは、計算機TX-0とTX-2を開発したウェス・クラークによってMITのリンカーン研究所で初めて提案されたパーソナル・コンピューターでありました。TX-0、TX-2はともに単一のユーザーにオンラインで使わせることを目的とした、非常に高価な機械でした。パーソナル・コンピューターは、1970年代に入る以前では経済的でありませんでしたし、多数の人々にオンライン・サービスを提供することと、少数の人々に高価なパーソナル・コンピューターを与えることとの間の競合は現在でも存在しています。

machines intended to be used on-line by a single user. Personal computers didn't become economical until the 1970s, and there is still a conflict between providing on-line service to many users and providing expensive personal computers to a few users.

Logic in Artificial Intelligence

The 1959 "Programs with Common Sense" paper said:

> The *advice taker* is a proposed program for solving problems by manipulating sentences in formal languages. The main difference between it and other programs or proposed programs for manipulating formal languages (the *Logic Theory Machine* of Newell, Simon and Shaw and the Geometry Program of Gelernter) is that in the previous programs the formal system was the subject matter but the heuristics were all embodied in the program. In this program the procedures will be described as much as possible in the language itself and, in particular, the heuristics are all so described.
>
> The main advantages we expect the *advice taker* to have is that its behavior will be improvable merely by making statements to it, telling it about its symbolic environment and what is wanted from it. To make these statements will require little if any knowledge of the program or the previous knowledge of the *advice taker*. One will be able to assume that the *advice taker* will have available to it a fairly wide class of immediate logical consequences of anything it is told and its previous knowledge. This property is expected to have much in common with what makes us describe certain humans as having *common sense*. We shall therefore say that *a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.*

The main reasons for using logical sentences extensively in A.I. are better understood by researchers today than in 1958. Expressing information in declarative sentences is far more flexible than expressing it in segments of a computer program or in tables. Sentences can be true in much wider contexts than specific

人工知能における論理

1959年の〝常識を持つプログラム〟の論文では、次のことを述べています。

「アドバイス・テーカーはある形式言語で書かれた文を操作することにより問題を解くために提案されたプログラムである。他のプログラムや形式言語を操作するために提案されたプログラム（ニューエル、サイモン、ショーの数理論理機械、およびゲレルンターの幾何のプログラム）と、アドバイス・テーカーとの違いは、前者が形式的体系が問題の主題であり、発見的手法はプログラム自身の中に埋め込まれているのに対して、後者では手続きはできるかぎりその言語自身の中で記述され、特に発見的手法はすべて言語自身の中で記述されていることである。

アドバイス・テーカーに期待されている主要な利点は、その振る舞いが、その記号的環境と、それに何が望まれているかということを物語る文を作ることによって、改良されうることにある。このような文を作るために、そのプログラムに関する知識、あるいはアドバイス・テーカーが以前に持っていた知識は、ほとんど要求されない。アドバイス・テーカーは、それに対して語りかけられたことが何であるにしても、それらとそれ以前の知識の論理的帰結の相当広い集合を使うことが可能である。この性質は常識を持っているものとし、ある人間を記述することにかなり共通のものを持っていることが期待されている。

したがって、我々は、『もし一つのプログラムがそれに語りかけられた事柄、およびすでにそのプログラムが知っている事柄からの帰結を十分幅広くそれ自身で自動的に演繹するならば、そのプログラムは常識を持っている』と主張するのである」

人工知能において論理式を広く使う主要な理由は、1958年のころよりも現在のほうが、研究者によって、より良く理解されていることです。情報を宣言的な文で表現することは、情報を計算機のプログラムの切片、または表で表現することよりもはるかに柔軟性を持っています。論理式は特定のプログラムが有用であることよりもはるかに広い前後関係において真になりうるものであります。事実の供給者は、受け取る側の関数を、また受け取る側がどのように使用するか、あるいはそれを使うかどうかについて、それほど理解している必要はありません。同一の事実は種々な目的に使われてよいのです。何故ならば事実の集まりからの論理的帰結が入手可能だからであります。

アドバイス・テーカーという趣意書は1958年当時大きな野望でありましたし、現在

programs can be useful. The supplier of a fact does not have to understand much about how the receiver functions or how or whether the receiver will use it. The same fact can be used for many purposes, because the logical consequences of collections of facts can be available.

The *advice taker* prospectus was ambitious in 1958, would be considered ambitious today, and is still far from being immediately realizable. This is especially true of the goal of expressing the heuristics guiding the search for a way to achieve the goal in the language itself.

The formalism given in "McCarthy 1959" was just a sketch of a theory of the achievement of goals by sequences of actions. It took a more definite form in the *situation calculus* of a 1964 Stanford report and was published in "McCarthy and Hayes 1969." The basic idea is to use the formula

$$s' = result(e,s)$$

to represent the new situation s' that results when the event e occurs in situation s. The events most studied are actions, and there are usually conditions that s has to satisfy before we can infer much about s. The situation calculus embodies a special case of reasoning about actions and other events. First, the events can be regarded as discrete; they occur in one situation and result in another, and we don't need to reason about what happens during the event. Second, we consider only one event occurring at a time; concurrent events are not analyzed.

One feature of situations was emphasized conceptually but didn't play much role in the actual axiomatizations. Situations were regarded as infinitely detailed, e.g. a block on the table was in a particular location and had a detailed shape and distribution of material, perhaps down to an atomic level. Therefore, the formalism did not provide for knowing a situation exactly but only for knowing facts about a situation that partially characterized it. In this respect situation calculus differed essentially from the mathematical model used in physics and discussed in most philosophy of science, e.g. in gravitational astronomy. In physics models, it is customary to decide what planets are to be taken into account and whether to represent them as mass points or whether to consider (say) some moments of their mass distributions. In contrast to this, situation calculus was intended to provide a model to which new detail could be added at any time. I contend that this open-endedness is an essential characteristic of the common sense information situation, whether the reasoning is done by people or by machines. I now

でもそのように考えられていますが、その実現からはまだ遠く離れています。その言語自身の中で、ゴールへ到達する道程を案内するような発見的手法を表現するという決勝点はまだ実現から程遠いのは確かであります。

(マッカーシー　1959) における定式化は、一連の行動によってゴールに到達するという理論の素描にすぎません。それは1964年のスタンフォード・レポートの情況計算法において、より明確となり、かつ (マッカーシーとヘイス　1969) に発表されました。その基本的なアイデアは、情況sにおいて出来事eが起こったときに情況s'を表現するのに、式

s'=result (e,s)

を用いることであります。出来事のうちでよく研究されているのは、行為の場合であります。そして、我々が情況s'を推論する前に情況sを満たすべき条件が存在するのが普通であります。情況計算法は行為、およびその他の出来事を推論することの特別の場合を具体化しています。第一に、出来事は不連続に起こることであると考えられます。出来事は一つの情況の下で起こり、次の情況へ結果をもたらし、出来事が起こっている間のことを推論する必要はありません。第二に、ある時刻に唯一の出来事が起こると考えます。同一時刻に共同に作用するような出来事は分析されません。

情況の一つの特徴は、概念的には強調されるけれども、実際の公理化にあたってはそれほど役割を持っていないことにあります。情況は、無限に詳しいものと考えられるものであって、例えばテーブルの上の一つの積木の例であれば、積木は特定の場所にあり、詳しい形の記述や原子段階に至るまでの材料の区分等々であります。したがって、定式化は情況の詳細を的確に知ることではなくて、情況について、それを部分的に特徴づける事実を知ることについてだけ行われます。この点に関しては、情況計算法は物理学で使われたり、また、例えば重力天文学などの科学哲学で議論されている数学的模型とは本質的に異なるものであります。物理学のモデルではどの天体を考慮に入れるか、あるいは、どれを質点として取り扱えばよいのか、質量分布のモーメントを考えればよいのかといった種々のことを決めるのが普通であります。これに対して、情況計算法は、どんなときにも新しい詳細をつけ加えることができるような模型を与えることを目指しています。私は、この開放性が、推論を人間が行うにしても、また機械が行うにしても、常識・情報・状況の本質的な特性であると主張するものであります。私は、情況のように無限な詳細をもつ存在を豊かな存在と呼び、これを完全に記述可能な不連続な存在と対照して考えたいと思います。推論は豊かな存在を近似するための不連続な存在を使うだけではなくて、推論の過程の間に近似のレベルを

refer to entities like situations that have infinite detail as rich entities and contrast them with discrete entities that can be completely described. Reasoning involves using discrete entities to approximate rich entities, but the level of approximation can change during a reasoning process.

The situation calculus was used in Cordell Green's (1969) Ph.D. thesis along with resolution theorem proving and a method of "answer extraction" he devised. However, he and his colleagues at SRI found their theorem prover did too much search to be practical. My opinion was this was because they didn't have any way of using heuristic facts to control the search, but I didn't have a proposal of how to do it. I have more ideas about that now, but they still don't amount to a definite proposal for controlling reasoning with facts.

In this event, Fikes and Nilsson (1971), went to a restricted formalism called STRIPS in which resolution theorem proving was used for reasoning about properties of single situations, whereas going from one situation to the next was done by a program that interpreted the action descriptions directly. The consequence was a faster program, but it didn't allow sentences that involved more than one situation. Because the control problem still isn't solved, the practical A. I. systems that reason about actions all use restricted languages.

In the late 1970s' the introduction of formalized nonmonotonic reasoning revolutionized the use of logic in A.I. (McCarthy 1977b, 1980, 1986), (Reiter 1980), (McDermott and Doyle 1980).Traditional logic is monotonic in the following sense. If a sentence p is inferred from a collection A of sentences, and B is a more inclusive set of sentences (symbolically $A \subset B$), then p can be inferred from B.

If the inference is a logical deduction, then exactly the same proof that proves p from A will serve as a proof from B. If the inference is model-theoretic, i.e. p is true in all models of A, then p will be true in all models of B, because the models of B will be a subset of the models of A. So, we see that the monotonic character of traditional logic doesn't depend on the details of the logical system but is quite fundamental.

While much human reasoning corresponds to that of traditional logic, some important human common sense reasoning is not monotonic. We reach conclusions from certain premisses that we would not reach if certain other sentences were included in our premisses. For example, learning that I own a car, you conclude that it is appropriate on a certain occasion to ask me for a ride, but when you learn the further fact that the car is in the garage being fixed you no longer draw that conclusion. Some people think it is possible to try to save

変えることができるものであります。

情況計算法は、レソリューションによる定理証明法と、彼が考案した『解答抽出法』を取り入れた方法でもって、コーデル・グリーン（1969）の学位論文で初めて使われました。しかし、彼とSRIにおける協力者たちは、彼らの定理証明システムが実用的であるとは言い難いほどの量の検索を行っていることに気づきました。私の意見としては、探索を制御するような発見的事実をまったく使わなかったということが理由であると考えていますが、しかし、それをどのように行うべきかの提案は持っていません。現在では私はもう少し多くのアイデアを持ってはおりますが、事実を使って推論を制御する確定的な提案をするにはまだ十分ではありません。

ついに、ファイクスとニルソン（1971）はSTRIPSと呼ばれる強化された定式化に到達しました。そこでは、レソリューションによる定理証明は単一の情況の性質に対する推論に使われ、一つの情況から次の情況へといくには行為の記述を直接解釈するプログラムを使っています。その帰結として、より速いプログラムが得られましたが、二つ以上の情況に関する論理式は許容しませんでした。制御の問題が現在でも未解決なので、行為に関して推理を行うような実用的な人工知能システムはすべて、制限された言語を使っています。

1970年の後半に至って、形式化された非単調推論の導入は人工知能における論理の使用を革命的なものにしました（マッカーシー　1977b、1980、1986；ロイター　1980；マックダーモットとロイド　1980参照）。在来の論理は次のような意味で単調であります。

式pが式の集合Aから推論されたときには、Aを含むような式の集合B（A⊂Bと記号で表す）からもpが推論される。

もし推論が論理的演繹であれば、AからpをBからの証明として役立つものであります。もし推論がモデル論的であるならば、すなわち、Aのすべてのモデルについてpが真であるならば、Bのすべてのモデルにおいてpは真になるはずであります。なぜならば、Bのモデルの集合は、Aのモデルの集合の部分集合になっているからであります。したがって、在来の論理の単調性は基本的なものであって、論理的体系の詳細には依存していません。

人間の推論の多くは在来の論理に対応して単調であるけれども、いくつかの重要な人間の常識的推論は非単調であります。前提から結論に到達することができ、しかし、他の式を前提に付加すると同じ結論を導くことができないことがあります。例えば、私が車を持っていることを貴方が知っていて、時には私の車に乗せてもらうことがで

monotonicity by saying that what was in your mind was not a general rule about asking for a ride from car owners but a probabilistic rule. So far it has not proved possible to try to work out the detailed epistemology of this approach, i.e. exactly what probabilistic sentences should be used. In fact, it seems that the probabilistic reason would end up using nonmonotonic techniques. Anyway, A.I. has moved to directly formalizing nonmonotonic logical reasoning.

Formalized nonmonotonic reasoning is under rapid development and many kinds of systems have been proposed. I shall concentrate on an approach called circumscription. It has met with wide acceptance and is currently the most actively pursued. The idea is to single out among the models of the collection of sentences being assumed some "preferred" or "standard" models. The preferred models are those that satisfy a certain minimum principle. What should be minimized is not yet decided in complete generality and may be problem dependent. However, many domains that have been studied yield quite general theories using minimizations of abnormality or of the set of some kind of entity. The idea is not completely unfamiliar. For example, Ockham's razor, "Do not multiply entities beyond necessity," leads to such minimum principles if one tries to formalize it in logic.

Minimization in logic is another example of an area of mathematics being discovered in connection with applications rather than via the normal internal development of mathematics. Of course, the reverse is happening on an even larger scale; many logical concepts developed for purely mathematical reasons turn out to have A.I. importance.

As a more concrete example of nonmonotonic reasoning, consider the conditions under which a boat may be used to cross a river. We all know of certain things that might be wrong with a boat, e.g. a leak, no oars or motor or sails, depending on what kind of a boat it is. It would be reasonably convenient to list some of them in a set of axioms. However, besides those we can expect to list in advance, human reasoning will admit still others, should they arise, but we cannot be expected to think of them in advance, e.g. a fence down in the middle of the river. This is handled using circumscription by minimizing the set of "things preventing the boat from crossing the river," i.e. the set of obstacles to be overcome. If the reasoner knows of none in a particular case, he or it will conjecture that the boat can be used, but if he learns of one, he will get a different result when he minimizes.

This illustrates the fact that non-monotonic reasoning is conjectural rather

きると結論しても、その車が実は故障を直してもらうために修繕屋に預けてあることを貴方が知れば、もはや同じ結論を得ることはできないのです。貴方の心にあったのは、車の所有者に車に乗せてもらうことを頼むための一般的な規則ではなくて、確率的規則であったのだと主張することによって、単調性を救うことができると考えている人々も存在するのです。この方法の詳細な認識論を研究すること、すなわち、どんな確率的文を使うべきかを明確にすることを試みて成功した人はいません。実際に確率的理由もまた非単調な手法に終わってしまうように思えるのです。いずれにしても人工知能の研究は、まっすぐに非単調な論理的推論の形式化へ向かって進んでいます。

非単調推論の形式化は、速やかな展開の下にあり、多くの種類のシステムが提案されてきています。私は、限界付け(circumscription)と呼ばれる方法に集中して話を進めようと思います。それは広く認められており、今も、最も活発に研究されているものであります。このアイデアは、仮定されている文のモデルの中から、〝望ましい〟、あるいは〝標準的な〟モデルを抜き出すためのものであります。〝望ましい〟モデルは、ある種の極小条件を満たすものであります。何を極小化すべきかについては、完全に一般的にはまだ決められておりませんし、問題によって異なるかもしれません。しかし、これまで研究されてきた多くの領域で、異常さの極小化、ないし、ある種類の存在の集合の極小化を使った極めて一般的な理論が得られています。この考え方は、まったく新しいものではなく、例えばオックハムの剃刀、〝必要な範囲を超えて実存を繁殖させてはならない〟は、これを論理の中で形式化することを試みれば、このような極小条件に到達します。

論理における極小化は、数学の正当的な内部発達からくるよりも、応用と関連して発見されつつある数学の一分野のもう一つの例であります。もちろんこの反対のこともさらに大きいスケールで起こりつつあります。純粋に数学的な理由のために開発された多くの論理的概念が人工知能に対して重要であることが判明しているのです。

非単調推論のもう少し具体的な例として、川を渡ることのできるボートの条件について考えてみましょう。我々は一つのボートについて何か悪い条件であるような事柄を知っています。例えば、水漏り、ボートの種類によってオールが無いか、エンジンが無いか、または帆が無いといったことであります。これらの事柄のいくつかを公理の集合として、リストアップすることはかなり便利でありましょう。しかし、あらかじめリストに取り入れることを期待できる事柄のほかに人間の推論は、さらに他の事柄も、もしそれが起きればそれを許容できるけれども、それらをあらかじめ考えることは期待できません。例えば、川の真ん中に柵が張られているなどです。こういった

than rigorous. Indeed it has been shown that certain mathematical logical systems cannot be rigorously extended, i.e. that they have a certain kind of completeness.

It is as misleading to conduct a discussion of this kind entirely without formulas as it would be to discuss the foundations of physics without formulas. Unfortunately, many people are unaware of this fact. Therefore, we present a formalization by Vladimir Lifschitz (1987) of a simple example called "The Yale shooting problem." Drew McDermott (1987), who has become discouraged about the use of logic in A.I. and especially about the non-monotonic formalisms, invented it as a challenge. (The formal part is only one page, however). Some earlier styles of axiomatizing facts about change didn't work right on this problem. Lifschitz's method works well here, but I think it will require further modification.

In an initial situation there is an unloaded gun and a person named Fred. The gun is loaded, then there is a wait, and then the gun is pointed at Fred and fired. The desired conclusion is the death of Fred. Informally, the rules are (1) that a living person remains alive until something happens to him, (2) that loading causes a gun to become loaded, (3) that a loaded gun remains loaded until something unloads it, (4) that shooting unloads a gun and (5) that shooting a loaded gun at a person kills him. We are intended to reason as follows. Fred will remain alive until the gun is fired, because nothing can be inferred to happen to him. The gun will remain loaded until it is fired, because nothing can be inferred to happen to it. Fred will then die when the gun is fired. The non-monotonic part of the reasoning is minimizing "the things that happen" or assuming that "nothing happens without a reason."

The logical sentences are intended to express the above 5 premisses, but they don't explicitly say that no other phenomenon occurs. For example, it isn't asserted that Fred isn't wearing a bulletproof vest, nor are any properties of bulletproof vests mentioned. Nevertheless, a human will conclude that unless some unmentioned aspect of the situation is present, Fred will die. The difficulty is that the sentences admit an *unintended minimal model,* to use the terminology of mathematical logic. Namely, it might happen for some unspecified reason the gun becomes unloaded during the wait, so that Fred remains alive. The way nonmonotonic formalisms, e.g. circumscription and Reiter's logic of defaults, were previously used to formulate the problem, minimizing "abnormality" results in two possiblities, not one. The unintended possibility is that the gun mysterious-

ことは、〝ボートが川を横切ることを妨害する事柄〟の集合、すなわち、克服すべき障害の集合を極小化することによる限界付けによって克服されます。もし推論者が特定の場合について何も知らなければ、そのボートを使うことができると推測を持つでありましょうし、他方、一つの障害を知っているならば、極小化したとき別な結論が得られるでしょう。

このことは、非単調推論が厳密であることよりも予測的である事実をよく説明しています。実際に、ある種の数理論理学的システムは厳密な意味では拡大できないことが示されています。すなわち、ある種の完全性をもつ場合であります。

式を用いないで物理学の基礎を議論することのように、上記のような種類の議論を式を用いないで行うことは誤りであります。不幸にして多くの人がこのことに気づいていません。したがって〝イェールの射撃問題〟と呼ばれる、簡単な例のブラディミール・リフシッツ（1987）による形式化を述べましょう。ドゥリュー・マクダーモット（1987）は、人工知能における論理の使用に対して落胆し、特に非単調の形式化について落胆した人であるが、彼は挑戦してこの問題を発明しました。（しかし、形式化した部分は一頁ですむ）変化に関して事実を公理化するというやや古い方式はこの問題に対しては役に立ちません。リフシッツの方法は、ここでうまく働くけれども、私はさらに一部変更することが必要であると考えています。

初期情況では、弾の込めていない鉄砲と、フレッドいう一人の人がいます。鉄砲が弾込めされると、しばらく待ってから、鉄砲はフレッドに照準を合わせ、そして、発射されます。そして、望ましい結論はフレッドが死ぬことであります。形式ばらないで言えば、規則は次の通りです。（1）一人の生きている人は彼に何かが起こらないかぎりそのまま生きている。（2）弾込めは、鉄砲に弾が込められていることの原因になる。（3）弾が込められている鉄砲は、何かが弾をぬくまで、弾込めされたままである。（4）射撃は鉄砲を弾込めされていない状態にする。（5）弾込めされている鉄砲を他人に向けて発射すれば、その人は死ぬ。我々は次のように推論するつもりである。フレッドは鉄砲が発射されるまでは生きている。何故ならば彼に何かが起こるようなことは何も推論できないからである。鉄砲はそれが発射されるまで弾込めされている。なぜなら鉄砲に何かが起こるとは推論できないからである。フレッドは鉄砲が発射されたときには死にます。この推論の非単調な部分は〝起こること〟を極小化する、あるいは、〝理由が無ければ何も起こらない〟と仮定することにあります。

論理式は上の5個の前提を表現するように意図されています。しかし、それらは、他の事象が起こらないということを陽に言ってはいないのです。例えば、フレッドは

ly becomes unloaded.

Lifschitz's Cauality Axioms for the Yale Shooting Problem

Lifschitz's axioms use the situation calculus but introduce a predicate *causes* as an undefined notion.

We quote from (Lifschitz 1987).

"Our axioms for the shooting problem can be classified into three groups. The first group describes the initial situation:

$$holds(alive,\ SO), \qquad (Y1.1)$$
$$\neg holds,(loaded,\ SO) \qquad (Y1.2)$$

The second group tells us how the fluents are affected by actions:

$$causes(load,\ loaded,\ true), \qquad (Y2.1)$$
$$causes(shoot,\ loaded,\ false), \qquad (Y2.2)$$
$$causes(shoot,\ alive,\ false). \qquad (Y2.3)$$

These axioms describe the effects of *successfully performed* actions; they do not say *when* an action can be successful. This information is supplied separately:

$$precond(loaded,shoot) \qquad (Y2.4)$$

The last group consists of two axioms of a more general nature. We use the abbreviations:

$$success(a,s) \equiv \mathrm{A}\, f(precond(f,a) \supset holds(f,s)),$$
$$affects(a,f,s) \equiv success(a,s) \wedge \exists v\ causes(a,f,v)$$

One axiom describes how the value of a fluent changes after an action affecting this fluent is carried out:

$$success(a,s) \wedge causes(a,f,v) \supset (holds(f,result(a,s)) \equiv v = (true). \qquad (Y3.1)$$

防弾チョッキを着ていないとか、あるいは防弾チョッキの性質についても何も言っていません。それにもかかわらず、人間は情況に述べられていないことが存在するのでなければ、フレッドは死ぬと結論するのです。困難な点は、これらの文が数理論理学的用語を使用すると、意図していない極小モデルを許容してしまうという点にあります。すなわち、しばらく待っている間にある明記されていない理由により鉄砲は弾込めされていない状態になり、フレッドは生きたままになってしまうことが起こりうるのです。非単調は形式化である限界付けや、またはライターの欠席裁判論理では、問題を上記のように定式化するけれども、〝異常な〟結果を極小化するのは二つの可能性についてであって、一つではないのです。意図されていない可能性というのは、しばらく待っている間に不可思議にも鉄砲が弾込めされていなくなることであります。

イェールの射撃問題のためのリフシッツの因果公理

リフシッツの公理は情況計算法を使うけれども、その他に定義を持たない記法として述語*causes*を導入しています。

以下は（リフシッツ　1987）からの引用である。

『射撃問題に対する我々の公理は三つのグループに分類される。第一のグループは初期情況に関する次のものである。

holds (*alive*, So), (Y 1, 1)

¬*holds* (*loaded*, So). (Y 1, 2)

第二のグループは我々に変数が行為によってどのように影響するかを教えてくれる。

causes (*load, loaded, true*), (Y 2, 1)

causes (*shoot, loaded, false*), (Y 2, 2)

causes (*shoot, alive, false*). (Y 2, 3)

これらの公理は行為が成功裡に実行されたときの効果を記述する公理である。しかし、それらはどんなときに行為が成功できるかについては何も主張していない。この情報は次のように別に与えられる。

precond (*loaded, shoot*). (Y 2, 4)

最後のグループはより一般的性質をもつ二つの公理から成る。我々は次のような略記法を用いることにする；

success (a,s) ≡V f (*precond* (f,a) ⊃ *holds* (f,s)),

affects (a,f,s) ≡ *success* (a,s) ∧∃v,*causes* (a,f,v).

第一の公理は変数の値が、この変数に影響を与える行為が実行された後、どのよう

(Recall that *v* can take on two values here, *true* and *false*; the equivalence in *Y3.1* reduces to *holds (f,result(a,s))* in the first case and to the negation of this formula in the second.) If the fluent is not affected then its value remains the same:

$$\neg affects(a,f,s) \supset (holds(f,result(a,s)) \equiv holds(f,s)). \qquad (Y3.2)''$$

Minimizing *causes* and *precond* makes the right thing happen. While these axioms and *circumscription policy* solve this problem, it remains to be seen whether we can write a large body of common sense knowledge in the formalism without getting other unpleasant surprises. Another current question is whether we can get by with axioms about the external world only or whether the axioms must contain information about the purposes of the reasoning in order to determine the preferred models. Moreover, there are many more possibilities to explore for the formal minimum principle required for common sense reasoning.

Conclusions and Remarks

1. Progress in using logic to express facts about the world has always been slow. Aristotle didn't invent any formalisms. Leibniz, who wanted to replace argument by calculation in human affairs, didn't invent propositional calculus, although it is technically far easier than the infinitesimal calculus of which he was a co-inventor with Newton. Boole, who invented propositional calculus, and who called his book, *"The Laws of Thought"* didn't invent predicate calculus. Frege and his successors saw no need or possibility of formalizing non-monotonic reasoning. It seems to me that almost any of these ideas would have been accepted by preceding innovators, e.g. Leibniz would have accepted propositional calculus, had it been suggested. Therefore, we must conclude that we humans find it difficult to formulate many facts about our thought processes that are apparent when suggested.

Even with formalized non-monotonic reasoning there are obstacles to expressing the general facts about the common sense world in an epistemologically adequate and elaboration tolerant way. I take this as a sign that major innovations are yet to come, and I will have some suggestions in my more technical lecture.

I have always wondered why science doesn't progress more rapidly than it

に変わるかを記述している。

success (a,s) ∧ *causes* (a,s,v)

⊃(*holds* (f,*result* (a,s)) ≡ V≡*true*). (Y 3, 1)

(vは二つの値*true*,*false*を変域としている。Y 3, 1における≡は、vが*true*のとき *holds* (f,*result* (a,s)) に帰着され、vが*false*のときこの式の否定に帰着される) もし変数が影響を受けないときにはその値は同じ値にとどまる。

affects (a,f,s) ⊃ (*holds* (f,*result* (a,s)) ≡ *holds* (f,s)), (Y 3, 2)』

*causes*と*precond*とに関して極小化を行えば、適切なことが起こる。これらの公理と限界付けの政策はこの問題を解いてしまうけれども、この形式化の下で常識的知識の大きな本体を他の不愉快な驚きを得ることなく我々が書き下すことができるか否かは疑問点として残ります。望ましいモデルを決定するために、外の世界だけに関する公理でよいのか、それとも、推論を行う目的に関する情報を公理が含まねばならないのかといったことが現在研究されつつある問題であります。さらに常識推論のために要求されている形式的な極小原理を探究する多くの可能性が存在しています。

結びと注意

1. 世界に関する事実を表現することに論理を使うことの進歩はいつも遅々としたものでした。アリストテレスは形式的方法を発明しませんでした。ライプニッツは人間に関する事柄についての議論を計算に置き換えることを望んだ人ですが、ニュートンとともに発明した無限小計算より技術的にはるかに容易であるにもかかわらず、命題計算を発明しませんでした。ブールは命題計算を発明し、彼の本に〝思考の法則〟と名づけたが、述語計算を発明しませんでした。フレッゲとその後継者たちは、非単調推論を形式化することの必要性も可能性も見い出さなかったのです。私には、これらのアイデアのいずれもが前段階の発見者にとって許容されるべきものであったと思えるのです。例えば、ライプニッツは、もしすでに提案されていたなら、命題計算を許容していたでありましょう。したがって、我々は、示唆されたときに明白である人間の思考過程に関する多くの事実を定式化することは、本当に難しいことであると結論しなければなりません。

   形式化された非単調推論についても、認識論的に適切であり、改良が容易であるように、常識的世界についての一般的な事実を表現することには、いくつかの障害が存在します。このことを、私は主要な革新がまだ起こっていないこ

does. Progress in some sciences is limited by instrumentation and experimental technique. For example, this is true of molecular biology. In other sciences, progress is limited by mistaken ideas that direct attention away from the ideas that lead to progress. When this occurs, it is often hard to say why a discovery wasn't made 20 or more years earlier than it was. For example, there is no technical reason why non-monotonic reasoning couldn't have been formalized in the 1920s.

2. Artificial intelligence research took form in the late 1950s in ways that were quite different from the ideas of senior scientists who considered the problem. I, and I think most other A.I. researchers avoided our seniors, rather than either following them or opposing them. This was possible, because it turned out that we were not dependent on them either for research support or academic position. I suspect this would not be good advice in general. However, it seems to have turned out well in artificial intelligence in that the ideas related to A.I. proposed by most of them—Von Neumann, Wiener and McCulloch have not so far been fruitful. In my opinion, only Turing proposed the line of research that has given us the main results in A.I. achieved so far. Of course, the older proposals have not been refuted, and maybe new people will find them fruitful. Connectionism might be regarded as one such gamble.

3. My work in A.I. has been in accordance with a certain philosophical point of view. To what extent the philosophy has influenced the research is harder to say. My subjective impression is philosophy has been important.

a. Our human knowledge of the world does not consist of knowledge of sense data or even summaries of it. Material objects and human purposes, for example, are far more stable than our sense impressions of them. This isn't peculiar to the human situation; it will also be true for any robots we might build. It is possible that babies learn from their experience that material objects exist, but it's also possible that this organization of the world is in our genes. Animals that presume material objects and some other things are likely to have an evolutionary advantage. From the A.I. point of view, this suggests that we build into our programs presumptions of structures that exist in the real world. If we are going to use logical sentences to represent knowledge, then there should be variables ranging over material objects, and also variables ranging over other entities like

との印であるととっており、これに対する私の示唆を私の専門的な講演で述べるつもりであります。

私はいつも、科学の進歩が実際よりもなぜもっと速やかに行われないのだろうかと疑問に思っています。いくつかの科学の領域では、器械の設置や実験の技法によって進歩が制限されています。例えば、分子生物学ではこのことは正しいでしょう。他の科学では、進歩につながるようなアイデアから離れた方向に注意を向けるような、誤ったアイデアによって進歩が制限されることもあります。このようなことが起こりますと、一つの発見がなぜそれが為されるよりも20年前、ないしそれ以上前に為されなかったかということは難しくなります。例えば非単調推論がなぜ1920年代に形式化されなかったかという技術的理由は無いのです。

2．人工知能の研究の1950年代にとった形は、そのころの古参の科学者たちの考えとはまったく異なっていました。私、および他の人工知能研究者達もそうだと私は思うのですが、先輩の科学者たちに従ったり反対したりするのでなく、むしろ避けたのでした。このことは可能でした。なぜなら我々は研究資金や大学における地位に関して彼らに従属していなかったからです。このような助言は一般的には良くないかもしれません。しかし、人工知能に関しては、諸先輩――フォン・ノイマン、ウィーナー、マッカロッチ――によって提案されたアイデアのいずれもが、これまでのところ実を結ぶものではなかったということがはっきりしてきたと思えるのです。私の意見としては、チューリングだけが、人工知能の研究がこれまで到達した主要な結果へ、我々を導いてくれるような研究の方向づけをした人でありました。もちろん、古い提案を否定するものでもありませんし、また新しい人々によって、実り多いものと再発見されるかもしれません。コネクショニズムはこうした意味での試みとも考えられます。

3．私の人工知能の研究成果は、ある種の哲学的な物の見方に従っています。しかしどの程度まで哲学が私の研究に影響したかを言うことは難しいと思います。私自身の主観的な印象では、哲学は重要でした。

a．我々人間の世界に関する知識は感覚データやその要約から成り立っているものではありません。物質的対象、あるいは、人間的目的といったものは、それらに対する我々の感覚的印象よりもはるかに安定したものであります。このことは人間の情況に対してだけのことではありません。このことは、我々が作ることができるであろうロボットにとっても当てはまるのであります。赤ん坊が物

actions and beliefs.

This may seem obvious, but it is in contrast with some views. For example, some people have proposed that since the experience of a human or robot is representable by a sequence of sense impressions, then intelligence might consist of the ability to predict the future of a sequence from its value up to a given point. This presumes nothing about the world and therefore might be preferred by an extremely cautious philosopher. Efforts to program computers to predict sequences have not been informative, nothing that wasn't obvious resulted from the experiments, and the experimenters were at a loss for how to proceed further.

I suppose the positivistic emphasis on sense data is a residue of the early 20th century reaction against 19th century idealistic philosophy. This philosophy involved many vague postulated entities, and the reaction against it took the form of admitting only the most directly observable entities as basic.

b. Our human ascription of mental qualities, e.g. beliefs, to each other is warranted by the usefulness of the ascriptions in understanding, predicting and affecting other people's behavior. We don't know whether the tendency to do this is genetic, but one explanation of autistic children might be a failure of such a mechanism to develop normally. It is legitimate to ascribe mental qualities to any system where it helps explain its behavior. This idea is developed as philosophy in "Dennett 1971 and 1987" and for A.I. in "Newell 1980" and "McCarthy 1979a." It still isn't generally accepted in philosophy.

c. Many qualities, including especially mental qualities, that we ascribe to various systems are meaningful only in terms of approximate theories that enable us to understand aspects of phenomena but which cannot be given precise definitions in terms of the state of the world. See (McCarthy 1979a).

d. Today artificial intelligence is far from being able to produce systems of human capability in general reasoning. This has led to a modesty that one might also recommend to philosophers. An A.I. system involving some concept like *belief* can't purport to use "the true" completely general concept of belief, because there is no agreement about that. Indeed at the level of precision required for computational implementation, there aren't even any candidates. Therefore, A.I. has to get by with limited, approximate concepts. But maybe that's all humans have either. Maybe the philosophers' attempts to understand belief in

質が存在することを経験から学ぶことも確かなことでありますが、しかし、世界の構成は我々の遺伝子によって存在することも確かなことであります。物質的対象やその他のことを想定する動物というのは、進化論的有利性を持っていると考えられます。人工知能の観点から言えば、このことは、実際の世界に存在する構造について想定されることを、プログラムの中に組み入れることを示唆しています。もし我々が知識を表現するために論理式を使おうとするならば、物質的対象を変域にもつ変数や、行動や信念といった他の存在物などを変域とする変数が存在しなければなりません。

このことは明白であると思われるかもしれませんが、しかし、それは他のいくつかの見方と対照的であります。例えば、人間あるいはロボットの経験は感覚的印象の列として表現することができるのでありますから、知能は、現時点までの値から、ある列の未来の価を予測する能力から成り立っていると提案する人達もいました。このことは、世界について何も仮定していないので、極めて注意深い哲学者がよくとる立場かもしれません。系列の未来を予測するようなプログラムを作る努力は有益ではありませんでした。この実験の結果から明白でないものが得られた試しはなく、実験者達はその後どのように実験を続けるべきかわからなくなったのです。

感覚データを実証論的に強調することは、19世紀の観念論的哲学に対する20世紀初頭の反動の剰余であると私には考えられます。前世紀の哲学は多くの漠然とした公準的存在を含んでおり、それに対する反応は、直接観察できる存在だけを基礎として許容する形をとったのでした。

b．精神的資質、例えば信念を、人間がお互いに帰することは理解、予測、他の人の振る舞いに影響を与えるものに帰することの有利性によって保証されています。このようにすることへの傾向が遺伝的であるか否かは知りませんけれども、自閉症の子供に関する一つの説明として、このような機構の正常な発育ということが言えるかもしれません。それ自身の振る舞いを説明するシステムに、精神的資質を帰することは適正なことであります。このアイデアは哲学では（デネット　1971，1981）、人工知能では（ニューウェル　1980）、（マッカーシー1979a）で展開されています。しかし、哲学ではまだ一般的に許容されているとはいえません。

c．我々が種々なシステムに帰する、とりわけ精神的資質を含む多くの資質は、現象の情況を理解させてくれるが、世界の状態に関しては正確な定義を与えるこ

general merely force them to construct a concept rather than discover it. Maybe there is no completely general concept of belief that corresponds to what humans actually do or what computers can be programmed to do.

References:

Dennett, D.C. (1971): "Intentional Systems," *Journal of Philosophy* vol. 68, No.4, Feb. 25.

Dennett, D.C. (1987): *The Intentional Stance,* MIT Press.

Fikes, R. and Nils Nilsson (1971): "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence,* Volume 2, Numbers 3,4, January, pp. 189-208.

Green. C, (1969): "Application of Theorem Proving to Problem Solving," in IJCAI-1, pp. 219-239.

Jeffress, Lloyd A. (ed.) (1951): *Cerebral Mechanisms in Behavior : The Hixon Symposium,* Wiley.

Lifschitz, Vladimir (1987): "Formal Theories of Action", in F.M Brown (ed.), *The Frame Problem in Artificial Intelligence,* Morgan Kaufmann, pp. 35-58. Reprinted in M.L Ginsberg(ed.),*Readings in Nonmonotonic Reasoning,* Morgan-Kaufmann, pp. 410-432.

McCarthy, John (1956): "The Inversion of Functions Defined by Turing Machines," in *Automata Studies, Annals of Mathematical Study, No. 34,* Princeton, pp.177-181.

McCarthy, John (1959): "Programs with Common Sense", in *Proceedings of the Teddington Conference on the Mechanization of Thought Processes,* Her Majesty's Stationery Office, London.

McCarthy, John and P.J.Hayes (1969): "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in D. Michie (ed), *Machine Intelligence 4,*

とができないような近似的理論によってのみ意味を持っているのであります。（マッカーシー　1979a）を参照してください。

d．今日人工知能は、一般的推論を行う人間の能力を持つようなシステムを作り上げることからは、まだ程遠いと言わざるをえません。このことは、哲学者に推薦して良いような謙遜さを生むものである。信念のような概念に関係する人工知能システムは、本当の意味で完全に一般な信念を意味することはできません。なぜなら本当の意味での信念については何らの合意もないからであります。実際に計算機にインプレメントするために必要なだけの精密さのレベルにおいて、候補者さえないのであります。したがって、人工知能は制限された近似的な概念しか持ちえません。もっとも、すべての人間も近似的にしかこうしたものを持っていないとも考えられます。哲学者が一般的に信念を理解しようとしている試みは、単に彼らをして概念を発見するのではなく、概念を作ることに努力させることになっているのかもしれません。人間が実際に行動すること、あるいは、計算機がそのようにするようプログラムされていることに対応する完全に一般な信念という概念は、存在しないのかもしれません。

American Elsevier, New York, NY.

McCarthy, John (1977a): "History of LISP," in Proceedings of the ACM Conference on the History of Programming Languages, Los Angeles.

McCarthy, John (1977b): "Epistemological Problems of Artificial Intelligence," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* MIT, Cambridge, Mass.

McCarthy, John (1979a): "Ascribing Mental Qualities to Machines," in *Philosophical Perspectives in Artificial Intelligence,* Ringle, Martin(ed.),Harvester Press, July 1979.

McCarthy, John (1979b): "First Order Theories of Individual Concepts and Propositions," in Michie, Donald (ed.) *Machine Intelligence 9,* University of Edinburgh Press, Edinburgh.

McCarthy, John (1980): "Circumscription-A Form of Non-Monotonic Reasoning," *Artificial Intelligence,* Volume 13, Numbers 1,2, April.

McCarthy, John (1982): "Common Business Communication Language," in *Textverarbeitung und Bürosysteme,* Albert Endres and Jürgen Reetz (eds), R. Oldenbourg Verlag, Munich and Vienna, 1982.

McCarthy, John (1983): "Some Expert Systems Need Common Sense," in *Computer Culture: The Scientific Intellectual and Social Impact of the Computer,* Heinz Pagels (ed.), vol. 426, Annals of the New York Academy of Sciences.

McCarthy, John (1986): "Applications of Circumscription to Formalizing Common Sense Knowledge," *Artificial Intelligence,* April 1986.

McCarthy, John (1987): Generality in Artificial Intelligence," *Communications of the ACM.* Vol.30, No.12, pp.1030-1035.

McCarthy, John (1987): "Mathematical Logic in Artificial Intelligence," in *Daedalus,* vol. 117, No.1, American Academy of Arts and Sciences, Winter 1988.

McDermott, D. and J. Doyle (1980): "Non-Monotonic Logic I," *Artificial Intelligence,* Vol.13, No.1.

Newell, Allen (1981): "The knowledge Level," *AI Magazine,* Vol.2, No.2.

Reiter, R.A. (1980): "A Logic for default reasoning," *Artificial Intelligence,* 13 (1, 2), pp. 81-132.

Robinson, J. Allen (1965): "A Machine-oriented Logic Based on the Resolution Principle," *JACM,* 12(1),pp. 23-41.

Shannon, Claude and John McCarthy (eds.) (1956): *Automata Studies,* Annals of Mathematics Study 34, Princeton University Press.

Turing, A.M. (1950): "Computing Machinery and Intelligence," *Mind* **59**, pp. 433-460.

稲盛財団1988——第4回京都賞と助成金

発　　行　1992年8月20日
発 行 所　財団法人稲盛財団
京都市下京区四条通室町東入函谷鉾町87番地　〒600
電話〔075〕255−2688
製　　作　㈱ウォーク
印刷・製本　大日本印刷株式会社
ISBN4-900663-04-2 C0000