The Computation in Game Playing is Beyond the Reach of Computers' Ability

Jianfeng Zhou
942 Hough Ave. #D
Lafayette, CA 94549
jianfz@aol.com

Abstract

In game playing, there exist two kinds of computations: qualitative computation and quantitative computation. The quantitative computation belongs to Turing computation, but the qualitative computation is not. The qualitative computation is a kind of computation based on inconsistent or contradictory properties. So there is an essential limitation for current computers to imitate humans' thinking in game playing, which is a typical intelligent problem solving example. Therefore, a new mathematical computational model is required for an artificial intelligent system.

Introduction

At present, it seems that game playing is not a proper dynamic system in which an intelligent agent can situate. It may be considered that all the moves on a game board are discrete and certain information, and there are not real continuity and uncertainties. As a matter of fact, the continuity and the uncertainties behind the moves of chessmen are not only real, but also related to intelligent activities of game players. All the certainties and uncertainties directly reflect the intelligent activities. So game playing is a typical intelligent problem solving system.

Computers seem to be able to compute almost anything if algorithms can be constructed for them. However, all kinds of algorithms are created by human beings. Can we give out an algorithm that can generate all kinds of procedures required by game playing? If not, can computers learn what algorithms they need in game playing? Yet knowing how to learn is as difficult as knowing how to play. Learning and problem solving are two aspects of an intelligent activity. So it is necessary to study the essence of both humans' game playing and computers to see whether there is any fundamental difference between them.

Computation in Game Playing

I have spent several months to specifically study the thinking processes of human beings in GO game playing. GO game is popular in Japan, Korea and China. It is the most complicated game in the world. It only consists of two stones: black and white. There is a very simple rule for stones to live on the game board. And two players alternate in placing black and white stones on a large (19" by 19" line) ruled board, with the aim of surrounding territory. One stone is put on a game board in one step. Stones are never moved after they are put on a game board, and only removed if they are completely surrounded.

During the game playing, the precise information to a player is the position of a stone on a game board. Behind each step, there exist a lot of undetermined information, such as what the purpose is, what the next possible step is, what the advantage or disadvantage is. The information is not only uncertain, but also dynamically changes because a player may change his/her ideas and utilize stones to achieve different purposes, or because of the shift between the influence and territory of stones.

There are also a set of concepts used to describe the uncertainties in GO game playing: these include influence, territory, attack strength, defensive ability, potential for attack, strategies, tactics and so forth. They define some kinds of relations among stones. There are some factors related to these concepts, but the relations are very fuzzy and non-linear. For example, there is a straight line of white stones on a game board. The longer the line is, the greater its influence is to the black stones. There is a common knowledge that tells how great the influence of a group of stones is. However, the knowledge is dependent on the situation around the stones.

When human beings play a game, they always carry out two kinds of computations. One is at a qualitative level. They never try to foolishly test all possibilities of chessmen positions to find out a good step. At this level, the target of computation is to find out where the most important place or problem is and how to attack the problem. Let us call it a qualitative computation. It works on all contradictory relations such as an attack-defense relation, an influence-territory relation, a life-death relation and so on. Some contradictory relations occur at any time. For example, an attack-defense relation applies to a single stone or a group of stones. In other words, any stone has both aggressive and defensive meanings. The qualitative computation is a spiral process. It includes two processes. One is to analyze existing problems and get out the primary problem. The second process is to analyze the primary problem and work out a good strategy. The problems in game playing are in fact the dialectic contradictions among different contradictory aspects such as attack and defense. It is those contradictory relations that drive qualitative computation into a spiral process.

The second computation in game playing occurs at a quantitative level. The aim of this kind of computation is to obtain the sequence of a set of moves in a scope on a board. The sequence at this level is so important that it may decide the "life" or "death" of stones. It considers every possible step. This kind of computation can be referred to as quantitative computation.

The two kinds of computations above are inter-dependent. The qualitative computation decides the targets and the strategies, and the quantitative computation gives out the exact steps. Without the qualitative computation, the quantitative computation may get into an explosive situation because of too many possibilities of chessmen playing. The qualitative computation actually gives out a scope for the quantitative computation. On the other hand, without the quantitative computation, the qualitative computation cannot be implemented because it is based the computation of qualitative properties. The quantitative computation is a computation no matter what kinds of quantities they are. But the qualitative computation concerns the meanings or importance of quantities. And the meanings of different quantities are normally contradictory. Different quantities represent different aspects of a problem such as attack and defense. And these aspects are inter-related and dialectically contradictory. The qualitative computation will find out the primary contradiction among different contradictions. So the quantitative computation is a kind of Turing computation, but the qualitative computation is not.

Limitations of Computers

Computation in computers belongs to Turing computation, which is a purely quantitative computation. It goes only at one level. In other words, computers treat the essence of different quantities as the same. They do not concern the meanings of quantities. And the meanings of quantities depend on their applications created by programmers. Although an application can be built for qualitative purpose, the qualitative computation becomes totally static after it is implemented because the meanings of variables have been predefined by the application. The meanings of variables cannot be the variables of an application. That means that all the variables are of one type. However, the qualitative computation requires the meanings of quantities as its variables.

Is it possible to establish a computer system that takes both quantities and their meanings as variables? This problem is related to the meanings of symbols (Harnad 1990). In a current symbol system or mathematical formal system (Mendelson 1964), all symbols have only extensional meanings. In order to implement qualitative computation, both intension and extension should be assigned to symbols (Zhou 1997a). The intension of a symbol represents its continuous meaning. Can a Turing machine implement the intension of symbols? Turing machines are discrete models of computation (Turing 1956). They are based on current symbol systems, the symbols of which have only extensional meanings. So Turing machines are not strong enough to support qualitative computation.

Turing machines are a theoretical computational model of computers. It can only be used to compute all things that can be predefined or described. There is no computational part of a Turing machine to tell it how to generate new information if it is necessary (Hopcroft and Ullman 1979). One reason is that Turing machines run only at a quantitative level. Take expert or knowledge systems as an example, knowledge, meta-knowledge, or meta-meta-knowledge can be constructed in a knowledge system. However, there always exist exceptions whatever how much meta knowledge is built in the system. That is because the domain knowledge cannot be represented as the variables in a knowledge system and understood by the knowledge system itself.

Turing computation is limited in a consistent world. All software systems on computers should be consistent. It seems there is no doubt of this. Does an inconsistent system make any sense? From a consistent and static view, it is meaningless. However, an intelligent system has got to be able to compute contradictory information. Without this ability, a computer system is impossible to interact with a dynamic world, which is contradictory. Game playing is a good example of this contradictory world. It always presents contradictory problems. How does a purely consistent computer system interact with this world? All the software systems running on computers are consistent and static systems because they cannot evolve themselves into next generation without humans' help. Neither can expert or knowledge systems on computers. So it is impossible for computers to imitate humans' intelligent thinking in game playing

A New Computational Model

A new computational model is required in order to implement both the qualitative computation and Turing computation.

The basic elements of a new computational model should include two sides: discreteness and continuity. At one hand, computation focuses on the continuous natures of things; at the other hand, computation is carried on discrete natures. This is a crucial point to an artificial intelligent system. Different from Turing machines, a new computational model should always present two kinds of results. At one level, it gives out the nature, scope and strategy of the computation. At another level, it produces concrete values.

Symbols of a new computational model should consist of both discrete and continuous properties. That will provide the foundation for both quantitative and qualitative computations.

A new computational model should be able to compute contradictions. Whenever it meets new contradictions, it can automatically generate a new set of information to compute. That is a key point for a computation to behave as intelligent, creative or dynamic. In game playing, problems faced by a player in each step are often contradictory. There may be several benefits, but only one you can get. There may be some worse defects in your part and some great benefits attractive in other parts. However, only one action you can do in each step. The benefits may disappear in your next turn. All these contradictory information should be computed dynamically.

In short, the new computational element, or each step in a new computational model, should include two aspects. It should be able to make use of contradictory information to locate or limit the scope of a problem. And it should finally reach to a consistent result, such as defeating the opponent in game playing. That means a new computational model should dynamically accept and analyze the contradictory information and work out its consistent answer promptly.

Conclusion

Although there are scholars who argues that intelligence includes some kind of computation that is not Turing computation (Penrose 1989), the Church- Turing Thesis still prevails in the field of artificial intelligence. One reason is that it seems impossible to work out other computational model, the computability of which is greater than Turing machines'. At present, everything is formalized in a discrete and consistent world. The dynamic world is, however, continuous, dynamic and contradictory. Intelligent computation has to be based on contradiction so that it can present a both discrete and continuous, both static and dynamic world. What I am working on is an intelligent computational model to solve this impasse (Zhou 1997b).

References

Harnad, S., 1990, The Symbol Grounding Problem, Physics, D42

Hopcroft, J.E. and Ullman, J.D., 1979, Introduction to Automata Theory, Languages, and Computation, Addison-Welsley Publishing Co.

Mendelson, E., 1964, Introduction to Mathematical Logic, D. Van Mostrand Co.

Penrose, R., 1989, The Emperor's New Mind, Oxford University Press, Oxford

Turing, A.M., 1956, Can a Machine Think? The World of Mathematics, TEMPUS,Vol.4

Zhou, J., 1997(a), An Intelligent Logic System, the Second China-Japan Symposium on Advanced Information Technology, HuangShan, China

Zhou, J., 1997(b), How to Extricate Artificial Intelligence from a Difficult Position, Global Chinese Computer Conference on Education , Guanzhou, China