Control, computer engineering and information science

APPLICATION OF CELLULAR AUTOMATON IN PRODUCTION EXPERT SYSTEM

A.K. Stoyanov, V.A. Panov

Tomsk Polytechnic University
E-mail: Stoj-ak@ad.cctpu.edu.ru

The possibility of applying cellular automaton in production expert systems has been studied. For industrial expert systems having knowledge bases of several thousands rules, using cellular automaton as inference machines opens the possibility to increase the operating efficiency with knowledge bases.

Development of intellectual software is a topical task for specialists in information technology field.

Classical patterns of such software are production expert systems widely used in practice. In the basis of these systems there is application of rules to work with facts from some knowledge domain [1, 2]. Rules and facts save knowledge of experts about knowledge domain and inference machine is applied for obtaining results. The machine operation essence is that it checks up the conditions of rules satisfiability under the presence of several facts values set at the current point of time. The obtained result is the consequence of applying the chain of rules which connects fact values describing initial situation with fact values from knowledge base unknown before.

One of the problems which the developers of the expert system are faced with is disproportionately large growth of a number of checks at increase of a number of rules in knowledge base. In industrial knowledge base where a number of productions exceeds several thousands the addition of a small quantity of rules causes significant additional computations for obtaining the desired result.

Typical mode used for solving the problems connected with large volume of computations is paralleling at computation. Among the systems supporting parallelism at computation, cellular automatons which are used for solving various practical problems attracted our attention [3, 4].

Cellular automaton represents a lattice. Each cell of it takes a state (value) from the specified set of states with certain periodicity. The lattice of automatons may be of various types differing both in dimension and cell shape. Laws of changing cells states represent a set of rules. Each cell determines its new state by the states of its close neighbors using rules from this set. At appropriate set of rules this simple operational mechanism supports extensive hierarchy of structures and phenomena.

Comparing operation of inference machine of expert system with evolution of cellular automaton some parallels are detected.

First. Both inference machine and cellular automaton use the rules of the form for their operation: ***if* A *then* B**, where **A** is the condition and **B** is the action. In expert system these rules form a content of knowledge base and in cellular automaton they are laws of its development and they are called selection rules.

Proposition depending on some values of facts is meant by condition **A** in the rule. Proposition of this rule may be either verbal or mathematical. Validity of the proposition is checked-out and the result of the check depends on the values of facts included in it at the moment of check-out, i.e. on current state of the system using these rules and facts. In the case of expert system the state is determined by a set of fact values of knowledge base used before or being used at the moment of check-out. In the case of cellular automaton its state is determined by a set of states of single cells.

Actions fulfilled at successful condition check results (they may be intermediate acting further as conditions and target completing system operation) are meant by the action **B**.

Second. The state of inference machine of expert system and cellular automaton are occasionally updated and they depend on preceding state. In other words, one may talk both on expert system and cellular automaton evolution.

Final result of expert system evolution is a set of facts values connected by a chain of applied rules. The result of cellular automaton evolution depends on its initial state and content of selection rules. This result represents a set of values of cell state appearing on cellular automaton field as a result of applying selection rules. The question occurs how the cellular automaton will evolve in case its rules of selection are the rules of knowledge base and states of its singular cells are the values of facts of expert system knowledge base. Is it possible to direct cellular automaton evolution so that its results are such states of cells which could allow talking about the ability of using cellular automaton in expert systems?

To obtain answers to these questions the model of formal knowledge base of expert system and the model of cellular automaton evolution were developed. On the basis of these models a software prototype of expert system shell was implemented. In this prototype cellular automaton in which rules from knowledge base serve as transition rules is applied as the inference machine.

Knowledge base formalization

Constructing expert system knowledge base two problems should be solved. The first one is filling the base with knowledge. The second one is reducing this knowledge to the form convenient for their use in cellular automaton.

In real situation filling the base with knowledge requires involvement of experts and engineers-cognitologists in work. The task is simplified for developing expert system shell. In this case it is possible to come to some formal rules without semantics of some field of knowledge but giving a chance to check operation of inference machine and other components of expert system.

Formalization of knowledge base rules at simulating is obvious. Any fact and its values in knowledge base may be presented in string type. In real situation lines have rather definite sense. But the most important thing for expert system operation is not the meaning of separate lines — fact values but their conformity to the rules applied to them. In knowledge base of real expert system there should be logically connected chains of fact values which are detected at system operation. Formal rule recording should take this circumstance into account. In other words, at expert system simulating it is possible to abstract from fact meaning and their values in the knowledge base but following certain requirements. Exactly these random lines — fact values used in the rules should connect several rules into the logical chains. I.e. one and the same line — value should be included both in condition of one rule and conclusion (action) of another one. And though semantics of fact value is lost in this case the main thing — their connection into logical chains remains. Chains may be of random length and their quantity may also be random. Single rules not included into any chain may be in knowledge base along with chains.

Rather definite content may be further juxtaposed to facts and rules from the model described above. Knowledge base should be converted to numerical type for being used in cellular automaton. For this purpose all facts, values and rules of the base should be replaced by their numerical identifiers. At such replacement any base may be simulated by three sets of integers. In one of them figures are numerical rule identifiers, in the second one they are fact identifiers and in the third set — fact value identifier.

The rules of such model has the form:

$$N_i: \text{if } M_r = L_p, \text{ then } M_n = L_k,$$

where N_i; M_r, M_n and L_p, L_k are the numbers of identifier sets respectively: rules, facts and their values.

Cellular automaton evolution

Without details of software implementation schematically cellular automaton evolution used as inference machine looks like the following. Torus obtained by joining opposite sides of cellular automaton lattice serves as evolution field. On this field of a part of cells the initial values coinciding with numerical identifiers of some fact values from base of knowledge are specified. These values make initial configuration of cellular automaton. Then the values coinciding with numerical identifiers of knowledge base rules are specified on the field of cellular automaton of other parts of cells. Thus, all the rules being in the base are activated. The rest cells are empty i.e. they have no numerical values. Properties of cells-facts and cells-rules are different. Cells-facts exist («live») for a certain specified time and after that numerical value is removed from them and they come to a rank of empty cells («die»). Cells-rules may check the values of cells of specified environment and also exist for specified time.

Let time required for single renewing states of all cells of automaton be understood under the evolution period. Let close neighbors of selected cell be understood under environment. I.e. those having any common side or contacting by angles (diagonal neighborhood). At each next period of evolution a cell-rule examines the area and generates a new fact corresponding to its rule conclusion at free space of the area if there are cells-facts in the area coinciding with sending. Age of cells-facts is checked as well at each period. If a cell is not demanded and its age exceeds a threshold age then it «dies» i. e. become free.

When the rule works (fact is generated) its age increases and cells with facts being sendings for rules become «immortal». The rule working certain times for the time of its existence generates similar rule at any free place of its area. If the rule during its life has never worked then it «dies» i. e. cell corresponding to it becomes free.

The mechanisms of «punishment» and «stimulation» described above should not provide the selection but of those rules for which cells-facts included in their conditions during evolution exist. In this case it does not matter how these cells-facts occurred on the field. They may have entered into initial configuration of automaton facts and may have occurred at automaton evolution. Thus, evolution should result in selection of those rules which are interconnected and correspond to initial configuration from the set of all rules of knowledge base.

It is easy to notice that evolution of the described cellular automaton may be referred to those evolutions which are used at investigations referring to the so-called «*artificial life*» [1].

Experimental check

To check the possibility of applying cellular automaton as an inference machine the knowledge base consisting of two files was formed besides the automaton itself: file of rules and file of facts (attributes) with their values. Formats of data storage in them had the following form:

facts — (attribute_name)=(value_1) or (value_2)...;

rules —

   one fact is in sending -

     if (attribute_name_$k1$)=(value_$kk1$), then

      (attribute_name_kM)=(value _ kkM);

   two and more facts in sending -

     if (attribute_name_$k1$)=(value_$kk1$) and

    (attribute_name _$k2$)=(value_$kk2$)... and

     (attribute_name _kN)=(value_kkN),

      then (*attribute_name_kM*)=(value_kkM).

Fragments of files containing facts and rules are given below.

Fragments of fact files of knowledge base:

...

$(J)= (J1)$ *or* $(J2)$;

$(K)= (K1)$ *or* $(K2)$ *or* $(K3)$;

...

Fragments of rule file of knowledge base:

...

if $(HA)=(HA1)$, *then* $(FA)=(FA1)$;

if $(I)= (I1)$, *then* $(J)=(J1)$;

...

In these fragments symbols in brackets before the equality sign are the names of some facts and symbols in brackets after equality sign are the values of these facts. Identifiers of 48 facts with total amount of their values equal to 116 were indicated for the experiment in fact file. File of rules contained 46 rules composing four logical chains and three rules not included in chains. Altogether are 49 rules.

In a separate file of initial configuration the fact values used for evolution start of cellular automaton were stored:

$(C)=(C3)$; $(H)=(H1)$; $(A)=(A1)$; $(A)=(A2)$;
$(K)=(K2)$; $(Q)=(Q2)$; $(EA)=(EA2)$; $(HA)=(HA1)$

At base loading, the rules, facts and meanings were assigned with identifiers used at automaton operation. Initial values of facts given above contain four values being starts of four logical chains formed by facts and rules of knowledge base. Three values are not included in any chain of knowledge base.

Selecting such evolution parameters as rule lifetime and square lattice dimension of cellular automaton we were guided by the following considerations.

Let us proceed from the fact that rules should be detected at evolution time (at evolution periods) the one at which cell moving at the average in one direction on torus surface returns into starting point where it had started to move. For this time the automaton surface should already be filled with cells occurred at interaction of moving cells-rules with cells-facts and repeated passes along the same motion path are almost impossible.

Let us determine how many periods at the average are required for one such pass. Let p_i be the probability of cell-rule move in one direction at i evolution period, λ_i is the cell shift at i period, and d is the average path for T evolution periods. It is obvious that $d=\sum_i p_i\lambda_i$. Let cell-rule overlap stochastically in any direction s available to it. There are $s=8$ of such directions in the case of rectangular lattice and Moore area. The probability of moving in one direction at large quantity of free cells on automaton field (at the beginning of evolution) may be taken as equal to $p_i=1/s$. A cell shifts each time to the adjacent position i. e. $\lambda_i=1$. Substituting selected values p_i and λ_i into expression for d it is possible to obtain the ratio $d=T/s$. Moving in one direction a cell appears in initial point passing the path $d=K$, where K is the dimension of square sizes of cellular automaton field. Therefore, we obtain that to return to initial point the $T=Ks$ of evolution periods are required.

Let us estimate the dimension of size K included into expression. Magnitude K should support the area sufficient for evolution of cellular automaton i.e. it is necessary to have $K^2>>(R+F)$, where F is the number of facts of initial configuration and R is the number of rules placed on automaton field. The case when each fact of initial configuration and each rule of knowledge base are repeated some times for higher operating efficiency at the initial point of time at placing on automaton field should be provided as well. It means that we should take $F=\alpha f$ where α is the repetition factor of quantity f of fact value being in the file of initial configuration. The same is true relative to the rules i. e. $R=\beta r$ where r is the number of rules being in the file of rules of knowledge base and β is the repetition factor r.

Number F of fact values can not exceed a number of rules R i. e. $F=\mu R$ where $\mu<1$. Usually a number of initial facts is considerably lower than a number of rules. Let us take $\mu\approx0,1$, then we can consider that $K^2>>R$. It is considered in engineering evaluations that inequality is fulfilled if we take $K^2\approx(3...5)R$. Selecting multiplier before R equal to 3 we obtain the ratio $K\approx(3\beta r)^{1/2}$.

For higher operating efficiency at the initial time point on cellular automaton field each fact of initial configuration was repeated 12 times (a?=12) and each rule of knowledge base was repeated 8 times ($\beta=12$). Taking into account the fact that $r=49$ we obtain $K\approx42$ for square lattice dimension. Value $K\approx45$ is taken in the experiment to make correction for presence of fact values of initial configuration on automaton field.

Rule lifetime should be sufficient for passing completely along torus surface and returning into initial point i. e. it should exceed value $T=Ks$. Taking into account selected $K\approx45$ and the fact that for Moore area we obtain $s=8$ that this time should not be less than 360 periods. It is taken equal to 400 periods in the experiment:

Evolution was implemented at the following conditions:

- cellular automaton topology is torus;
- field dimension is 45×45;
- rule lifetime is 400 (periods);
- fact lifetime is 600 (periods);
- at rule response its lifetime increases up to 10 periods;
- threshold of rule generation is 5 responses;
- arrangement of facts of initial configuration and rules is random;
- direction of cell shift and position of produced result are random.

The dependence of cell relative quantity on automaton field occupied with rules and facts (dashed line) and relative quantity of activated rules from the total amount of those included into knowledge base (solid line) on time is given in Figure. It is not difficult to notice that a number of activated rules is almost permanent at the level of 0,6 from the total amount of rules starting from about the time amounting to 0,7 from rule lifetime. A part of occupied cells by this time amounts to more than 0,8 from their total amount on automaton surface and

rate of increment of this part decreases distinctly approaching to 1 i.e. complete surface filling.
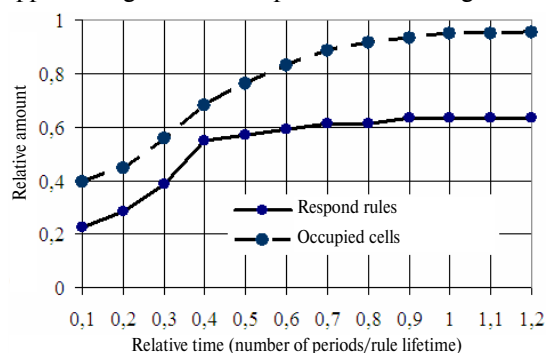

Figure. *Change of number of rules and cells at cellular automaton evolution*

The analysis of the list of rules activated in relative time equal to 1 showed that all existing chains were recovered in it completely or partially. The list of the rules included into logic chain recovered completely for this period is given below.

if $(A)=(A1)$ and $(C)=(C3)$ then $(B)=(B1)$;

if $(B)=(B1)$ then $(C)=(C1)$;

if $(C)=(C1)$ then $(D)=(D1)$;

if $(D)=(D1)$ then $(E)=(E1)$;

if $(E)=(E1)$ then $(F)=(F1)$;

if $(F)=(F1)$ then $(G)=(G1)$.

Only values $(A)=(A1)$ and $(C)=(C3)$ were in initial configuration of facts of this chain, the rest six are the result of cellular automaton evolution.

On the basis of the analysis of figure curve trend one may conclude that all chains may be completely recovered after repetition of automaton evolution. Really if evolution is stopped at relative time value equal to 0,6 then about 60 % of the rules are recovered. Repeated evolution may be carried out for the same (or a bit more) interval. But the values of facts of initial configuration should be changed. The values of conclusions of all chains final rules recovered at the first stage of cellular automaton evolution should be taken as such values. To check out this assumption one more evolution stage was carried out for the case when the logic chain given below was recovered. This evolution stage as a previous one was carried out for relative time equal to one. Values of facts included into modified file of initial configuration are given below:

$(M)=(M1)$; $(OA)=(OA2)$; $(I)=(I2)$; $(G)=(G1)$; $(N)=(N3)$; $(FA)=(FA1)$; $(ZA)=(ZA1)$; $(L)=(L2)$.

Let us notice that the given sequence contains fact values which may be called «noise». These are values included into single rules and rules situated at the end of logic chains detected at the first stage of evolution. Their presence is unavoidable as it is known beforehand whether they are included into chains or not. As a result of the second stage of evolution all logic chains were completely recovered in the studied cases for the rules of knowledge base.

Discussion of results and conclusions

The analysis of final state of cellular automaton evolution allows making conclusion that automaton really implemented the procedure of direct inference using rules of knowledge base as transition rules. In this case in rather adverse conditions (one rule of a chain) more than 60 % of rules included into logic chains of knowledge base were recovered. Increasing a number of facts of initial configuration, changing the parameters of cellular automaton evolution all logic chains may be completely recovered for rules of knowledge base of expert system.

Cellular automaton may be used developing expert systems the knowledge base of which consists of a great number of rules (units of thousands and more). Knowledge bases of industrial expert systems are the same. Approximate calculation disclaiming special accuracy shows that the use of cellular automaton as a inference machine at large quantity of rules may be more profitable than application of known strategies of inference in expert systems.

Really, let r of rules making a certain number of logic chains average length of each is L be saved in a base. In general expert system $r/2$ of condition checks are required at the average for detecting one rule of a chain and obviously $N_0 \approx Lr/2$ of checks for detecting the whole chain of rules.

Let us estimate a number of checks which is required of cellular automaton for solving the same problem − searching for rules of one logic chain. According to curve trend in the Figure it follows that about a half of rules are recovered for a half of lifetime of the rules at automaton evolution i.e. two stages of evolution with total duration equal one is required for recovering all chains. We determined earlier that lifetime of rules T is specified by the ratio $T=Ks$. Therefore, the total number of checks N carried out for time T for $R=b?r$ of cell-rules placed at the beginning of evolution on automaton field equals to $N=TRs$, or taking into account expression for T, $N=Ks^2R$. A number of chains with average length L may be taken equal to r/L i.e. $N_1=NL/r=Ks^2L\beta$ of checks are required for recovering one chain. Taking into account the fact that $K\approx(3\beta r)^{1/2}$ we obtain that a number of checks for one chain $N_1\approx(3r)^{1/2}s^2L(\beta)^{3/2}$.

Let us compare values N_1 and N_0. Their ratio N_1/N_0 gives value $2(3)^{1/2}s^2r^{-1/2}(\beta)^{3/2}$ and it is less than 1 at $R>12s^4\beta^3$. Estimation of value R for Moore area $s=8$ and for $\beta=1$ gives value $R>50000$. For the area of von Neumann $s=4$ and relative estimation amounts to $R>3000$. Therefore, at existence of such number of rules in knowledge base the use of cellular automaton allows achieving the result for fewer checks than at direct search of rules. Let us notice the fact that cellular automaton represents parallel computing environment is not taken into account in this estimation. Account of this fact should decrease this value.

Another opportunity opening when using cellular automaton is a preliminary detection of logic chains in expert system knowledge base for their further storage in

it and application at result obtaining. Such approach solves the problem of rapid preparation of agenda (chained list of rules ready for being fulfilled).

And, finally, preliminary detection of logic chains by cellular automaton reduces time for checking out consistency of system of rules when adding new rules into knowledge base. Really, the added rule if it is not includ-

ed into already stored logic chain may be consistently connected both to beginnings of one or several chains and to the end of some chain. Additional checks are not required.

Thus, cellular automaton application as inference machines opens capacity of increasing efficiency of work with bases of knowledge of industrial expert systems.

REFERENCES

1. Tim Johns M. Programming artificial intelligence in applications. Translated from English. – Moscow: DMK Press, 2004. – 311 p.

2 Gavrilova T.A., Khoroshevskiy V.F. Base of knowledge of intelligence systems. – St.-Petersburg: Piter, 2000. – 382 p.

3. Toffoly T., Margolus N. Machines of cellular automatons: Translated form English. – Moscow: Mir, 1991. – 282 p.

4. Loskutov A.Yu., Mikhailov A.S. Introduction into synergy. – Moscow: Nauka, 1990. – 270 p.

STUDYING THE SYSTEM OF PIPE HEADER TELEMECHANICS ON THE BASIS OF COMMUNICATION NETWORK OF GSM STANDARD

D.D. Zykov, A.A. Shelupanov

Tomsk state university of control systems and radioelectronics
E-mail: zdd@udcs.ru

Studying the system of pipe header telemechanics on the basis of communication network of GSM standard has been covered, the results of experiments for two services of GSM have been given. The way of increasing system response speed is suggested and GSM is compared with other services applied at the present in pipe header telemechanics systems.

Different companies all over the world propose solutions on the basis of GSM for tasks of accounting energy resources, navigation, telemetry, logistics, safety etc. [1–3].

Let us enumerate the main services of network of standard GSM: SMS, mode Data call and mode GPRS [4].

SMS (Short Message Service) finds wide application in industrial systems owing to low cost, simplicity in use and convenient service. The disadvantage of this service is limitation in length of massage (not more than 140 bytes). SMS are convenient at small volume of transferred data. For example, electricity meter may be interrogated once in a day.

Mode Data call is identical in its characteristics to general voice mode but in this case not coded voice signal is transferred but user data. Maximal transfer rate (which is constant during the whole communication session) equals to 9600 byte/s.

GPRS is the technology of batch communication (General Package Radio Service) in networks of GSM. The feature of this mode is in the fact that after connection payment is actually transferred information content is paid but not the time of connection.

Let us examine the system of telemechanics of pipe header linear part. It includes several control points (CP) set at pipeline in each 10...15 km and one control station (CS), Fig. 1.
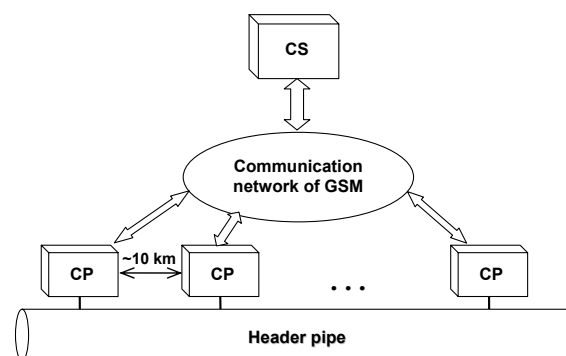

Fig. 1. *Diagram of the system of telemechanics using communication network of GSM standard*

In telemechanic systems (TS) all or some kinds of control information may be transmitted. Transferring information only on values of object parameters the TS is called the system of telemetering system (TM); in remote signaling system (RS) the information on the possible state (usually of two ones) of the controlled object is mainly transferred; in telecontrol system (TS) only control commands are transformed. In combined TS information of two types is transffered, for example, measuring and signaling (TM – TS), control and signaling (TC – TS). In complex TS control information of all types (TC – TS – TM) may be transffered.