An Anticipatory Reasoning Engine for Anticipatory Reasoning-Reacting Systems

Shinsuke NARA, Feng SHANG, Takashi OMI,
Yuichi GOTO and Jingde CHENG
Graduate School of Science and Engineering Saitama University,
255 Shimo-Okubo, Sakura-ku, Saitama, 338-8570, Japan
{nara, frank, omi, gotoh, cheng}@aise.ics.saitama-u.ac.jp

Abstract An anticipatory reasoning-reacting system (ARRS) has been proposed as a highly reliable and highly secure reactive system. The most important component of an ARRS is its anticipatory reasoning engine (ARE). We have proposed temporal relevant logic (TRL) as a sound logical basis of anticipatory reasoning, and shown that parallel processing techniques are effective to efficient anticipatory reasoning. This paper presents a real ARE we are developing based on TRLs. We define basic requirements of an ARE, discuss implementation issues for an ARE, present our implementation techniques, and show and discuss some current experimental results obtained by using our ARE. Our ARE can also be used in other computing anticipatory systems where anticipatory reasoning plays a key role.

Keywords : Anticipatory reasoning, Prediction, Temporal relevant logic, Forward deduction

1 Introduction

An anticipatory reasoning-reacting system (ARRS for short) has been proposed as a highly reliable and highly secure reactive system [9, 10, 13]. An ARRS is a computing system containing a controller C with capabilities to measure and monitor the behavior of the whole system, a traditional reactive system RS, a predictive model PM of RS and its external environment, and an anticipatory reasoning engine ARE such that according to predictions by ARE based on PM, C can order and control RS to carry out some operations with a high priority [9, 10].

An ARE is the most important component in an ARRS and it may be an application-independent general one [9, 10]. In order to implement an ARE, there are two issues; a sound logical basis underlying anticipatory reasoning are required [10]; anticipatory reasoning gets enough effective conclusions anticipatorily within an acceptable time in order to satisfy the requirement of high reliability and high security from applications [11]. We have proposed temporal relevant logics as a sound logical basis of anticipatory reasoning [10], been developing an automated forward deduction system for general-purpose entailment calculus (EnCal for short) which is one of a reasoning engine [5] and shown that parallel processing techniques are effective to efficient anticipatory reasoning [11] however there is no real ARE. We expect that we can develop an ARE by integrating them.

This paper presents a prototype implementation of an ARE. We explain related works of an ARE, show basic requirements which we have defined, discuss implementation issues, present our implementation techniques, show current experimental results and discuss the experimental results. We also show that Our ARE can also be used in other computing anticipatory systems where anticipatory reasoning plays a key role.

2 Relate Work

We summarize related works of an ARE. Anticipatory reasoning is reasoning to draw new, previously unknown and/or unrecognized conclusions about some future event or events whose occurrence and truth are uncertain at the point of time when the reasoning is being performed [10]. An ARE is an computing program which executes anticipatory reasoning.

Now we have no real ARE yet. We have only defined requirements of an ARE and we have only proposed some useful tools and techniques for developing an ARE. The requirements are as follows.

1. An ARE must deduce conclusions about some future event or events.
2. Conclusions deduced by an ARE must be correct if premises are correct.
3. Anticipatory reasoning gets enough effective conclusions anticipatorily within an acceptable time in order to satisfy the requirement of high reliability and high security from applications

From requirement 1 and 2, an ARE requires a logic system underlying anticipatory reasoning adequately. We have proposed four requirements which the logic system underlying anticipatory reasoning satisfies at least. The requirements are as follows; the logic must be able to underlie relevant reasoning as well as truth-preserving reasoning, it must be able to underlie ampliative reasoning, it must be able to underlie paracomplete and paraconsistent reasoning and it must be able to underlie temporal reasoning [10]. We also have proposed temporal relevant logic (TRL for short) which satisfy the requirements [6, 7, 10]. TRLs are obtained by introducing temporal operators and related axiom schemata and inference rules into strong relevant logics. TRLs are extensions of strong relevant logics [3, 4, 8].

From requirement 2, An ARE must be based on automated forward deduction because the conclusion deduced by it must be definitely correct if premises are correct [11]. Automated forward deduction is a process of deducing new conclusions from premises automatically by applying inference rules to the premises and previously deduced conclusions repeatedly until some previously specified termination conditions are satisfied [5].

As a forward deduction engine, we have been developing EnCal [5]. EnCal is a general forward deduction engine however we can not use EnCal as an ARE. EnCal leaves TRLs out of consideration.

From requirement 3, we have shown that parallel processing techniques are effective for efficient anticipatory reasoning. The techniques are the parallelization of automated forward deduction. We have implemented parallelized versions of EnCal according to the techniques on a shared memory parallel computer and a cluster of PCs. In both implementation, Speed-up ratio of forward deduction engine executing with 16 processors achieve almost 12 times to forward deduction engine executing with 1 processor by the parallel processing techniques [11]. This fact does not mean that we have implemented a parallelized ARE because parallelized versions of EnCal also leaves TRLs out of consideration.

3 Implementation

3.1 Prototype of an Anticipatory Reasoning Engine

An ARE is a forward deduction engine based on TRLs. An ARE makes candidates of predictions according to predictive model(PM) with data given by controller(C) and logical theorems of TRLs. We expect that we can develop an ARE by integrating them.

As a first step of developing an ARE, we implement a prototype of an ARE by simply improving EnCal. Our ARE take empirical theorems which are PM and data given by C and logical theorems of TRLs, inference rules and termination conditions as input. It deduces conclusions which are candidates of predictions form premises automatically by apply inference rules to the premises and previously deduced conclusions until some previously specified termination conditions are satisfied. If the termination conditions are satisfied then it outputs all candidates of predictions.

3.2 Implementation issues

EnCal is a general forward deduction engine however we can not use present EnCal as an ARE. The reasons are as follows.

1. Present EnCal can not deal with temporal operators defined in TRLs.
2. present EnCal can not deal with some inference rules defined in TRLs.
3. The termination condition of present EnCal is not suitable for an ARE.

First, EnCal can not deal with temporal operators such as until operator($\boldsymbol{U}$), since operator($\boldsymbol{S}$), future-tense sometime operator($\boldsymbol{F}$), past-tense sometime operator($\boldsymbol{P}$), future-tense always operator($\boldsymbol{G}$), past-tense always operator($\boldsymbol{H}$), tomorrow operator($\boldsymbol{T}$) and yesterday operator($\boldsymbol{Y}$) which are defined in TRLs [14]. EnCal only provides logical operators such as →, ↔, ⇒, ⇔, ∨, ∧ and ¬ which are defined in such as classical mathematical logics, relevant logics [1, 2] and strong relevant logics.

Second, EnCal can not deal with inference rules which is defined in temporal relevant logics too. EnCal provides only one inference rule whose name is modus ponens(from A and A⇒B to infer B) however we want to use other inference rules such as adjunction (from A and B to infer A∨B) and temporal generalization(from A to infer ***G***A and ***H***A) defined in TRLs and we also want to use some inference rules which is useful for anticipatory reasoning. We also select one or more inference rules in the inference rules because we do not always want to use all the inference rules.

Lastly, the termination condition of EnCal is not suitable for an ARE. In general, both the logical theorems of a logic system and empirical theorems deduced based on the logic from some premises are infinite sets of formulas, even though the premises are finite. In order to develop a computational tool for reasoning about logical and empirical entailment, Cheng constraints the conclusions to a set of formulas with low (entailment) degree [5]. This strategy can be applied to computational tools that reason about strong relevant logics. EnCal adopts a termination condition as a method proposed by Cheng [5]. The method is making a fragment of all conclusions deduced from premises. By the method deduced conclusions are limited by measuring degree of nested a implication which is a one of logical operators. The method can be used as a termination condition for a EnCal [5] however it can not work well as a termination condition for an ARE. In temporal relevant logics, there is a temporal generalization which is an inference rule joining temporal monadic operators ***G*** and ***H*** to a formula. If the ARE does not have the limit method for joining temporal monadic operators ***G*** and ***H*** to a formula, then the ARE may not terminate its execution.

3.3 Implementation of an ARE

For developing an ARE, we improve EnCal and adopt new termination condition.

First, we increase the number of operators which EnCal provides in order to deal with temporal operators. We add two binary operators which denote ***U*** and ***S*** and six monadic operators which denote ***G***, ***H***, ***F***, ***P***, ***T***, and ***Y*** to EnCal.

Second, we implement some inference rules which are adjunction, temporal generalization and some other inference rules which are useful for anticipatory reasoning. We can also select one or more inference rules which we want to use in the inference rules.

Lastly, we adopt new termination conditions for an ARE. We have proposed new limit method for TRLs [14]. We have introduced temporal degree of a formula which is nesting depth of temporal operators in that formula. Temporal degree(D_t) of a formula can be formally defined as:

1. $D_t(A)=0$ if and only if there is no temporal operator in A;

2. If A has the form of $\Psi(B,C)$, where Ψ is one of binary temporal operator, then $D_t(A)$=max($D_t(B)$;$D_t(C)$) + 1;
3. If A has the form of ΦB, where Φ is one of unary temporal operator, then $D_t(A)$=$D_t(B)$ + 1;
4. If A has the form of ϕB, where ϕ is one of unary logical connectives, then $D_t(A)$=$D_t(B)$
5. If A has the form of $B\phi C$, where ϕ is one of binary logical connectives, then $D_t(A)$=max($D_t(B)$;$D_t(C)$)
6. If A has the form of $\sigma x B$, where σ is one of quantifiers, then $D_t(A)$=$D_t(B)$ If $D_t(A) =$ i where i is a natural number, A is called a i^{th} temporal degree formula.

Let $(F(L), \vdash_L, Th(L))$ be one of TRLs, and k be a natural number. The k^{th} temporal degree fragment of L, denoted by $Th'^k(L)$, is a set of logical theorems of L which is inductively defined as follows (in the terms of Hilbert style formal system):

1. if A is an axiom of L, then $A \in Th'^k(L)$
2. if A is a $j^{th}(j < k)$ degree formula which is the result of applying an inference rule of L to some members of $Th'^k(L)$, then $A \in Th'^k(L)$.
3. Nothing else are members of $Th'^k(L)$.

Obviously, the definition of the k^{th} temporal degree fragment of logic L is constructive. Let $(F(L), \vdash_L, Th(L))$ be one of TRLs, premise $P \subset F(L)$, and k and j be two natural numbers. A formula A is said to be *j^{th}-temporal-degree-deducible* from P based on $Th'^k(L)$ if and only if there is an finite sequence of formulas $f_1 \ldots f_n$ such that $f_n = A$ and for all $i (i < n)$ $1) f_i \in Th'^k(L)$, or $2) f_i \in P$ or $3) f_i$ whose temporal degree is not higher than j is the result of applying an inference rule to some members $f_{j1} \ldots f_{jm}$ $(j1 \ldots jm < i)$ of the sequence.

If $P \neq \phi$, then the set of all formulas which are *j^{th}-temporal-degree-deducible* from P based on $Th'^k(L)$ is called the j^{th} temporal degree fragment with premises P based on $Th'^k(L)$, denoted by $T^j_{Th'^k(L)}(P)$. To carry out anticipatory reasoning in ARE, we can combine this strategy based on temporal degree with the strategy based on (entailment) degree in order to further narrow down the searching space of possible candidates for predictions.

4 Some Current Experimental Results

In order to show that our ARE works correctly, we experiment with our ARE under a scenario; a fire breaks out in a building which has ten floors and fire starts on a sixth floor. We make a simple PM which models behavior of fire as follows.

1. If a floor starts burning, then the fire will spread all over the floor, i.e., the floor becomes all burnt, and also spread upward and downward in different speeds.
2. if a floor is all burnt, then the floor does not burn again.
3. if a floor starts burning, then the floor must be going to be all burnt.
4. Time for a floor to be all burnt from start burning time is $\boldsymbol{T}$, time for fire to spread upper adjacent floor is also $\boldsymbol{T}$, and time for fire to spread lower adjacent floor is $2\times\boldsymbol{T}$.

If candidates of predictions deduced by our ARE correspond to the PM then our ARE works correctly, because An ARE makes candidates of predictions according to a PM.

We also make empirical theorems given by a C to an ARE. The empirical theorems is floor state. A floor has three states. We describe the states by predicates as follows; a predicate "NB(x)" denotes "x-th floor is not burning", a predicate "SB(x)" denotes "x-th floor is start burning" and a predicate "AB(x)" denotes "x-th floor is all burnt". The empirical theorems which the C gives to an ARE are as follows; NB(1), NB(2), NB(3), NB(4), NB(5), SB(6), NB(7), NB(9), NB(9) and NB(10). the PM is also described by formulas which are constructed by the predicates and operators of TRL. The number of formulas of empirical theorems given by the C and the PM are 84.

We show inference rules and axiom schemata used in the experiment. We use two inference rules. One is "$A, A\Rightarrow B \vdash B$" which is modus ponens. The other is "$A, B, (A\lor B)\Rightarrow C \vdash C$" which is short circuit version of modus ponens and adjunction. We use ten axiom schemata of TRL. These axiom schemata are follows; $((A\Rightarrow B)\Rightarrow((B\Rightarrow C)\Rightarrow(A\Rightarrow C)))$, $((A\Rightarrow \neg B)\Rightarrow(B\Rightarrow \neg A))$, $(\boldsymbol{G}(A\Rightarrow B)\Rightarrow(\boldsymbol{G}A\Rightarrow \boldsymbol{G}B))$, $(\boldsymbol{G}A\Rightarrow \boldsymbol{G}\boldsymbol{G}A)$, $(\boldsymbol{G}A\Rightarrow \boldsymbol{T}A)$, $\boldsymbol{G}(\boldsymbol{T}(A\Rightarrow B)\Rightarrow(\boldsymbol{T}A\Rightarrow \boldsymbol{T}B))$, $(\boldsymbol{T}(A\Rightarrow B)\Rightarrow(\boldsymbol{T}A\Rightarrow \boldsymbol{T}B))$, $(\boldsymbol{Y}(A\Rightarrow B)\Rightarrow(\boldsymbol{Y}A\Rightarrow \boldsymbol{Y}B))$, $(\boldsymbol{T}(A\lor B)\Rightarrow(\boldsymbol{T}A\lor \boldsymbol{T}B))$ and $(\boldsymbol{Y}(A\lor B)\Rightarrow(\boldsymbol{Y}A\lor \boldsymbol{Y}B))$.

We also show entailment degree and temporal degree. In the experiment, we specify entailment degree 2 and specify temporal degree 1 to 5.

Our test platform is a computer with 3GHz Pentium 4 CPU, 2G byte main memory. The table 1 shows the experimental results of our ARE. In the table, "D_t" denotes temporal degree, "conclusions" denotes the number of conclusions deduced by our ARE, "time" denotes a execution time, "s" denotes second, "up" denotes a conclusion which denotes a floor upper than 6th floor will start burning at the farthest future. "down" denotes a conclusion which denotes a floor lower than 6th floor will start burning at the farthest future. "burnt" denotes a conclusion which denotes a floor will be burnt at the farthest future. In the table 1, we use an abbreviation which is $\boldsymbol{T}_n$. $\boldsymbol{T}_n$ denotes a sequence of **n** characters of $\boldsymbol{T}$, i.e., $\boldsymbol{T}_4$ denotes $\boldsymbol{TTTT}$.

Table 1: The relation between temporal degree and conclusions deduced by our ARE

D_t	conclusions	time	up	down	burnt
1	402	3s	$\boldsymbol{T}_1$SB(7)	N/A	$\boldsymbol{T}_1$AB(6)
2	1445	39s	$\boldsymbol{T}_2$SB(8)	$\boldsymbol{T}_2$SB(5)	$\boldsymbol{T}_2$AB(7)
3	3645	261s	$\boldsymbol{T}_3$SB(9)	N/A	$\boldsymbol{T}_3$AB(5), $\boldsymbol{T}_3$AB(8)
4	7797	1261s	$\boldsymbol{T}_4$SB(10)	$\boldsymbol{T}_4$SB(4)	$\boldsymbol{T}_4$AB(9)
5	13861	4347s	N/A	N/A	$\boldsymbol{T}_5$AB(4), $\boldsymbol{T}_5$AB(10)

Conclusions which are deduced at upper temporal degree contain in conclusions which are deduced at lower temporal degree. In the table 1, we found following.

1. In conclusions whose temporal degree is $n(n \leq 5)$, $T_nAB(5+n)$ is included.
2. In conclusions whose temporal degree is $n(n \leq 4)$, $T_nSB(6+n)$ is included.
3. In conclusions whose temporal degree are 2, $T_2SB(5)$ is included and In conclusions whose temporal degree are 4, $T_4SB(4)$ is included.
4. In conclusions whose temporal degree are 2, $T_3AB(5)$ is included and In conclusions whose temporal degree are 4, $T_5AB(4)$ is included.
5. The number of conclusions becomes lager as temporal degree becomes higher.
6. The execution time of an ARE becomes longer as temporal degree becomes higher.

From result 1, 2, 3 and 4, we found that our ARE works correctly because our ARE makes candidates of predictions correctly according to our simple PM. We found that our ARE makes the farthest future candidates of predictions as temporal degree becomes higher too. From result 5 and 6, we found that if we want to make the farthest future candidates of predictions then an ARE must deduces more number of conclusions and must need more time.

5 Discussion

Through this implementation, we found new requirements for ARE. An ARE should accept premises which a C gives in process of a execution of an ARE. ARRSs are real time systems. Behaviors of the systems and its external environment must change with time however our ARE may not correspond to the changes. Our ARE only takes premises which the C gives as initial input. It can not accept newer premises until it finishes its execution therefore it can not make newer candidates of predictions. An ARE also should remove premises and/or conclusions which are deduce by the ARE. Behaviors of the systems and its external environment must change with time therefore premises and conclusions deduced by an ARE may become useless to make candidates of predictions. From these reasons, an ARE should be able to remove its premises and/or conclusions.

To develop a practical ARE, the most important issue is to improve an efficiency of ARE. In the experiments, we made a very simple PM and very small number of premises which a C gives however the execution time of our ARE is large when we specify temporal degree are 5. In practical use, a execution time of ARE must become larger because a PM becomes more complex and the number of premises which a C gives becomes larger. We have to implement an parallelization version of an ARE according to a parallel processing techniques which we have proposed. We also evaluate the techniques adequate for an ARE. If not, then we must propose new techniques for efficient anticipatory reasoning.

In practical use, we have a new issue. The issue is how we find useful, important and/or interesting conclusions in a lot of conclusions. We can not use useful, important and/or interesting predictions unless we found the predictions in conclusions deduced by an ARE even if the ARE deduces such predictions. In practical use, a number of conclusions must be large. As a number of conclusions deduced by an ARE becomes larger, investigating useful important and/or interesting conclusions becomes more difficult and its execution time becomes longer. We must automate the investigating process. We also propose some criterion in order to automate the investigating process.

Our ARE can be used by other computing anticipatory systems from two reasons. One is that our ARE is not specialized for an ARRS. The other is that an anticipatory system is one in which present change of state depends upon future circumstance rather than merely on the present or past [12]. In order to make future circumstance, our ARE can be used.

6 Concluding Remarks

In this paper, we have presented our prototype of an ARE and we have shown some current experimental results of our anticipatory reasoning engine. From the experimental results, we showed that our ARE is useful tool for anticipatory systems which requires predictions. From the experimental results, we also showed that our anticipatory reasoning engine is not practical one yet because the execution time of our anticipatory reasoning engine is large. Applications systems which use anticipatory reasoning engine can not allow the execution time. From discussion, we have proposed some new requirements for an anticipatory reasoning engine and some issues to develop a practical one. From the issues, as future works, we implement a parallel anticipatory reasoning engine with parallel processing techniques which we have shown in order to shorten the execution time and we also implement a new ARE which satisfies the new requirements.

References

[1] Anderson, A.R., Belnap Jr., N.D.: Entailment: The Logic of Relevance and Necessity, vol. 1. Princeton University Press (1975)

[2] Anderson, A.R., Belnap Jr., N.D., Dunn, J.M.: Entailment: The Logic of Relevance and Necessity, vol. 2. Princeton University Press (1992)

[3] Cheng, J.: Logical tool of knowledge engineering: Using entailment logic rather than mathematical logic. In: Proc. ACM 19th Annual Computer Science Conference, San Antonio, USA (1991) 228–238

[4] Cheng, J.: The fundamental role of entailment in knowledge representation and reasoning. Journal of Computing and Information **2** (1996) 853–873

[5] Cheng, J.: Encal: An automated forward deduction system for general–purpose entailment calculus. In: Terashima, N., Altman, E., (eds.): Advanced IT Tools, IFIP World Conference on IT Tools, IFIP96 – 14th World Computer Congress. Chapman & Hall (1996) 507–517

[6] Cheng, J.: Temporal relevant logic as the logic basis for reasoning about dynamics of concurrent systems. In: Proc. 1998 IEEE-SMC Annual International Conference on Systems, Man and Cybernetics, San Diego, California,USA (1998) 794–799

[7] Cheng, J.: Temporal relevant logic: What is and why study it? In: Abstract of the IUHPS/DLMPS 11th International Congress of Logic, Methodology and Philosophy of Science. (1999) 253

[8] Cheng, J.: A strong relevant logic model of epistemic processes in scientific discovery. In: Kawaguchi, E., Kangassalo, H., Jaakkola, H., Hamid, I.A., (eds.): Information Modelling and Knowledge Bases XI. IOS Press (2000) 136–159

[9] Cheng, J.: Anticipatory reasoning-reacting systems. In: Proc. International Conference on Systems,Development and Self-organization (ICSDS'2002), Beijing, China (2002) 161–165

[10] Cheng, J.: Temporal relevant logic as the logical basis of anticipatory reasoning-reacting systems. In: Dubois, D.M., (ed.): Computing Anticipatory Systems: CASYS 2003 - Sixth International Conference. AIP Conference Proceedings (2004) 362–375

[11] Goto, Y., Nara, S., Cheng, J.: Efficient anticipatory reasoning for anticipatory systems with requirements of high reliability and high security. In: Dubois, D.M., (ed.): International Journal of Computing Anticipatory Systems. Volume 14., CHAOS (2004) 156–171

[12] Rosen, R.: Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations. Pergamon Press (1985)

[13] Shang, F., Cheng, J.: Towards implementation of anticipatory reasoning-reacting system. In: Dubois, D.M., (ed.): International Journal of Computing Anticipatory Systems. Volume 14., CHAOS (2004) 93–109

[14] Shang, F., Cheng, J.: Anticipatory agents based on anticipatory reasoning. In: Hacid, M.S., Murray, N., Ras, Z.W., Tsumoto, S., (eds.): Proceedings of Foundations of Intelligent Systems, 15th International Symposium, ISMIS 2005. Volume 3488 of Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)., Springel-Verlag (2005) 445 – 455