Layered Argumentation For Fuzzy Automation Controllers

Insu Song
School of Business and IT
James Cook University Australia
Singapore
insu.song@jcu.edu.sg

Guido Governatori
Education in Queensland Research Laboratory
NICTA (National ICT Australia)
Australia
guido.governatori@nicta.uq.edu.au

Joachim Diederich
School of Business and IT
James Cook University Australia
Singapore
joachim.diederich@jcu.edu.sg

***Abstract*—We develop a layered argumentation system (LAS) for efficient implementation of Fuzzy automation controllers. LAS extends a logic based proposal of argumentation with subsumption concept and varying degree of confidences in beliefs. We show that this argumentation system can be used to model Fuzzy automation controllers. The argumentation system is based on a nonmonotonic logic, the computational complexity of which is known to be linear to the size of the knowledge base. LAS theories can also be mapped into RTL-VHDL (Register Transfer Level–VLSI Hardware Description Language) or RTL Verilog for very efficient hardware implementation of Fuzzy automation controllers.**

***Keywords*-Fuzzy logic, Fuzzy controller, Nonmonotonic logic, Argumentation system.**

I. Introduction

Since the late 1960's, Programmable Logic Controllers (PLCs) have been the most popular automation controllers despite the fact that most PLCs are based on a simple extension of propositional logic called *ladder logic*. This is the proof that a simple propositional logic (extended with counters, registers, and timers) can be successfully used to manage modern automation systems. The main reasons for this popularity are *robustness*, *reactiveness*, and *simplicity*. Robustness and reactiveness are particularly important if we want to control critical systems, such as nuclear power stations, 100 ton earthmoving machines, or mobile robots deployed in a distant planet. In these systems, system failures or any slight delay of responses must be avoided.

However, the applicability of PLC is still very limited to relatively simple processes mainly because it is difficult to manage even moderate sized ladder logic programs. It is unimaginable to program with ladder logic when we have tens of thousands of rules. In addition, despite its simplicity, most PLCs are still too big to be used for small stand-alone devices or too slow for certain applications. Therefore, PLCs (and ladder logic) are not suitable for low profile, high performance systems.

Therefore, we need to develop a more expressive (yet still simple) language that can be used to create small, robust, and reactive systems with interesting complex behaviors. One promising candidate language is a logic-based agent model because logical languages are close to our natural languages and many agent models are suitable for specifying complex autonomous behaviors that must reason with incomplete knowledge.

However, existing logical approaches [1], [2], [3], [4] suffer from some major shortcomings to be embedded in small low powered devices:

1) They have high computational complexity required for drawing conclusions;
2) Most logical formalisms cannot express varying degrees of belief which is essential for designing intelligent automation controllers;
3) They have difficulties in expressing behaviors;
4) They are neither reactive nor suitable for mission critical applications as they require sophisticated theorem provers running on a high powered CPU.

We solve these problems by:

1) Devising a layered argumentation system that can be computed efficiently or mapped into a combinational logic circuit on an electronic chip, such as FPGAs and ASICs;
2) Allowing expressions of relative certainties of rules and evidence similarly to Fuzzy-logic; and
3) Allowing decomposition of systems into parallel interacting behaviors for incremental construction of systems similarly to the subsumption architecture [5].

The benefits of this approach are – the resulting system on a chip is very small and reactive because it can compute all of the conclusions instantaneously regardless of the size of the agent's knowledge base (KB) and without using real-number for fuzzy computation; and it is robust because it neither rely on a CPU nor an operating system (i.e., there can be no fatal error messages or halts).

The paper is structured as follows. In the next section we illustrate a simple fuzzy-temperature controller, which we use to compare conventional fuzzy-logic approach with LAS in Section VI. In Section III, we then illustrate how a layered structure can be used to implement various controllers having complex behaviors. In Section IV, then we formally define LAS. After that, in Section V fuzzy argumentation operators are defined. Then, in Section VI we compare conventional fuzzy-logic approach with LAS using the simple fuzzy-temperature controller. In Section VIII we discuss our results

978-1-4244-6502-6/10/$26.00

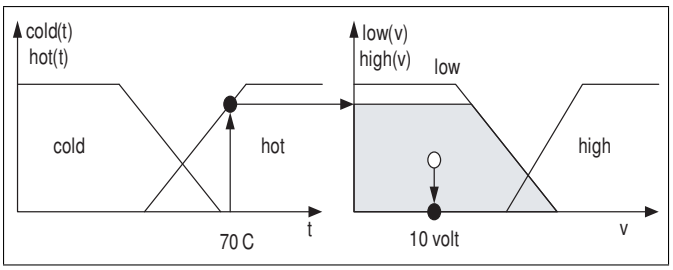


Fig. 1. Fuzzy inference process of a temperature controller.

with existing methods. In Section IX we conclude this paper with some remarks.

II. Example Fuzzy Controller

In this paper, we use a simple fuzzy controller to demonstrate and compare our new method with conventional Fuzzy-logic methods. The target system is a simple temperature controller described by the following two statements:

1) If water temperature is cold ($cold$), set voltage applied to the heater high ($high$).
2) If water temperature is hot (hot), set voltage low (low).

Suppose also that we are given the following fuzzy membership functions:

$$cold(t),\ hot(t),\ high(v),\ low(v)$$

A fuzzy membership function defines the membership of the input x to its class c: $c(X) : X \rightarrow [0, 1]$ where 0 is for absolute nonmembership of x and 1 is for absolute membership. The following is one possible output configuration of the fuzzy temperature-controller in Fuzzy-logic:

$$centroid(max(min(cold(t), high(v)), min(hot(t), low(v))))$$

where for *fuzzy-inference* we combined two membership functions with $min()$ and for *aggregation* we combined the inference results with $max()$. To obtain a crisp output value, we used centroid calculation for defuzzification[1]. Figure 1 illustrates how membership functions are used to infer output 10 volt given input temperature $70°C$. The left-hand side of the graph plots fuzzy-membership functions for $cold$ and hot temperature. The right-hand side plots fuzzy-membership functions for low and $high$ voltage. It illustrates an inference process given that the water temperature is $70°C$.. The output voltage after defuzzification with centroid calculation is 10 volt.

We show how this fuzzy controller can be implemented using our Layered Argumentation System in Section VI.

III. Behavior Based Decomposition of Agent Systems

In [5], Rodney Brooks introduced a subsumption architecture that decomposes a system into several layers of (possibly prioritized) parallel behaviors of increasing degrees of competence rather than the standard functional decomposition. The basic idea is that it is easier to model a complex behavior by gradually implementing from less competent sub-behaviors to more competent sub-behaviors. In addition, the relation between layers is that more competent layers subsume less competent layers. However, the original subsumption architecture does not scale well for non-physical systems and to overcome the limitation, we need to develop a logic based subsumption architecture [6]. To do this, we need to introduce a varying degree of confidence and a subsumption architecture into a logical system.

To illustrate the difference between behavior based decomposition and functional decomposition, let us consider a cleaning robot. A typical functional decomposition of this system might resemble the sequence:

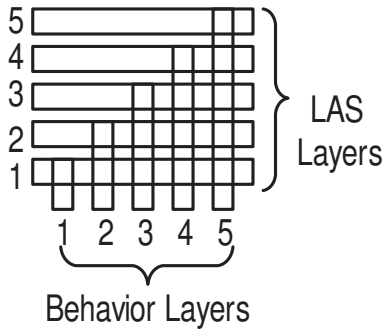
sensors $\rightarrow$ perception $\rightarrow$ modelling $\rightarrow$ planning $\rightarrow$ task selection $\rightarrow$ motor control.

The decomposition of the same system in terms of behaviors would yield the following set of parallel behaviors:

avoid objects $<$ avoid water $<$ clean $<$ wander $<$ map area

where $<$ denotes increasing degrees of competence. However, less competent layers are usually given higher priorities (i.e., given more confidence) than more competent layers. That is, decisions made by less competent behaviors usually override decisions made by more competent behaviors (more task specific behaviors). The reason is because less competent behaviors are usually more reactive and model more urgent behaviors. For example, avoiding an object is not much related to a specific task, but requires a quick response without much reconsideration because otherwise it will be too late.

However, this is not always the case because a strong 'will' can suppress a reactive behavior. Therefore, since the layers of LAS represent the confidence degree (certainty degree of information) instead of competence (task specificity), the layers of the subsumption architecture [5] do not exactly correspond to the layers of LAS as shown below.



In this figure, the bottom layer (layer-1) is the most confident LAS layer whereas the right most behavior layer (layer-5) is the most competent layer. It shows that less competent behaviors tend to be related with more confident layers whereas more competent behaviors tend to spread over several layers of LAS. As we will see in the following sections, the subsumption concept of [5] is used in decomposing concepts as well, because a less confident knowledge subsumes more confident knowledge by rule subsumption in LAS.

[1]Referring to the process of mapping each fuzzy inference-result (e.g. combined membership function) to crisp output value (e.g., voltage).

Example 1: Let us consider an example specification of a cleaning robot. Suppose that the robot performs cleaning (c) if it detects that the area is dirty (d); one option to clean the area is to turn on a vacuuming unit (v), but it must turn-off the vacuuming unit if it detects the presence of water (w). That is, we have two parallel behaviors: cleaning and avoiding water. However, avoiding water has higher priority than cleaning with a vacuuming unit. This specification can be represented as a set of inference rules decomposed into two layers of rules as follows:

$R_1 = \{r_1 : d \rightarrow c, \quad r_2 : w \rightarrow \neg v\}$

$R_2 = \{r_3 : c \rightarrow v\}$

The arrows herein represent defeasible inferences. For instance, $c \rightarrow v$ is read as "if c is true, then usually v is true". The layers represent relative confidences such that layer-n conclusions are more confident than layer-$(n+1)$ conclusions. Now, suppose the area is both dirty and wet with layer-1 confidence, then the vacuuming unit will not be turned-on: v is not true. The reason is because then $\neg v$ is a layer-1 conclusion by r_2, which prevents inferring layer-2 conclusion of v by r_3 and r_1.

We define fuzzy-argumentation operators in Section V and compare LAS with Fuzzy-logic for designing fuzzy controllers in VI.

IV. Definition of Layered Argumentation System (LAS)

In this section, we formally define a rule-based language called Layered Argumentation System (LAS). LAS is a logical language that can capture complex behaviors and varying degrees of agents' belief. LAS also offers Fuzzy-logic like reasoning features. A precursor of this language was used in [7] to decompose system behaviors similarly to Brook's subsumption architecture. Later in [8], it is given argumentation semantics and comparisons with other existing layered logics and hierarchical approaches.

A. Formal Definition

As the underlying logical language, we start with essentially propositional inference rules: $r : L \rightarrow l$ where r is a unique label, L is a finite set of literals, and l is a literal. If l is a literal, $\sim l$ is its complement: if l is a positive literal p, $\sim l$ is $\neg p$; if l is a negative literal $\neg p$, $\sim l$ is p.

An LAS theory is a structure $T = (R, N)$ where $R = (R_1, ..., R_n, ..., R_N)$ is a sequence of finite sets of rules where each R_n $(1 \leq n \leq N)$ is a finite set of layer-n rules and N is the number of layers and n is a layer index.

All rules in one and the same layer have the same degree of confidence. We stipulate that layer-n rules are more confident rules than layer-$(n+1)$ rules. This is because, when we build a system, we tend to add less task specific rules (i.e., more urgent behaviors) first, such as rules for avoiding objects in a corridor, and then gradually add more task specific rules, such as rules for finding a goal location. Importantly, LAS represents only the relative degree of confidence between layers in order to avoid the need for acquiring the actual confidence-degree value of every rule.

We will sometimes use layer index n (just) to index a certain imaginary confidence-degree value (*the degree of confidence*) of layer-n: the degree of confidence of layer index n is higher than the degree of confidence of layer index $(n+1)$. That is, layer-1 rules are the most confident rules and thus have the highest degree of confidence.

As for the semantics of the language, we modify the argumentation framework given in [9] to introduce layers into the argumentation system. Argumentation systems usually contain the following basic elements: an underlying logical language, and the definitions of: *argument*, *conflict between arguments*, and the *status of arguments*.

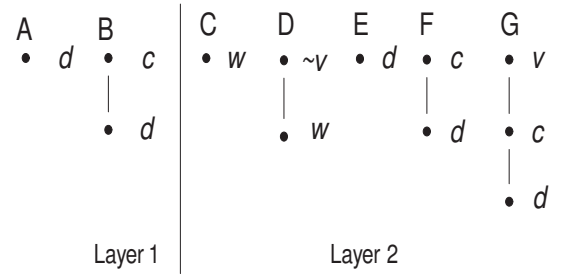
As usual, *arguments* are defined to be proof trees. An argument for a literal p based on a set of rules R is a (possibly infinite) tree with nodes labelled by literals such that the root is labelled by p and for every node with label h:

1) If $b_1, ..., b_i$ label the children of h then there is a rule in R with body $b_1, ..., b_i$ and head h.
2) The arcs in a proof tree are labelled by the rules used to obtain them.

A literal labelling a node of an argument A is called a conclusion of A. However, when we refer to the conclusion of an argument, we refer to the literal labelling the root of the argument. The set $Args_n$ of layer-n *defeasible arguments* is the set of arguments based on a set of rules $R_1 \cup ... \cup R_n$. All of arguments in $Args_n$ have the same layer index n: they have the same degree of confidence as layer-n. We define $Args_0$ to be the empty set. $Args_n$ is analogous to the defeasible arguments in [9]. The set $SArgs_n$ of layer-n *strict arguments* is a subset of arguments formed based on a set of rules $R_1 \cup ... \cup R_{n-1}$. All of arguments in $SArgs_n$ have the same layer index $n-1$: they have the same degree of confidence as layer-$(n-1)$. We define $SArgs_0$ to be the empty set. $SArgs_n$ is analogous to the strict arguments formed of the strict rules in [9]. That is, $SArgs_n$ is thought of as arguments that have already been demonstrated to be justified at layer-$(n-1)$.

We now introduce a set of usual notions for argumentation system: *attack*, *support*, and *undercut* at layer-n. A layer-n argument A *attacks* a layer-n argument B if the conclusion p of A is the complement of a conclusion q of B, and that conclusion of B is not part of a strict subargument C of B (i.e., $C \notin SArgs_n$). A layer-n defeasible argument A is *supported* by a set of layer-n arguments $S \subseteq Args_n \cup SArgs_n$ if every proper sub-argument of A is in S. A layer-n defeasible argument A is *undercut* by a set of layer-n arguments $S \subseteq Args_n \cup SArgs_n$ if S supports a layer-n argument B attacking a proper non-strict sub-argument C of A (i.e., $C \notin SArgs_n$).

Example 2: Consider the theory given in Example 1 with the following assumptions (arguments) added: $R_1 = \{\rightarrow d\}$ and $R_2 = \{\rightarrow w\}$. Now we consider the arguments below:



A and B are layer-1 defeasible arguments for d and c, respectively, and thus they are also a layer-2 defeasible arguments (E,F) because layer-1 arguments are more confident than layer-2 arguments. C, D, and G are layer-2 defeasible argument for w, $\sim v$, and v, respectively. F is a sub-argument of G. We should note that D is a layer-2 defeasible argument, because layer-1 rule r_2 is subsumed to layer-2. That is, layer-2 evidence for w and layer-1 rule r_2 are combined to produce a layer-2 argument. In addition, since there is no layer-1 evidence for w, there is no layer-1 argument for $\sim v$. We should note that this feature of combining evidence and rules of different layers is not possible in existing layered approaches such as [10]. Unlike Fuzzy Logic approaches [3], [11], the inference process is very simple and more intuitive.

The heart of an argumentation semantics is the notion of an *acceptable argument*. Based on this concept it is possible to define *justified arguments* and *justified conclusions*, conclusions that may be drawn even taking conflicts into account. Given a layer-n argument A and a set S of layer-n arguments (to be thought of as arguments that have already been demonstrated to be justified), we assume the existence of the concept: A is *acceptable w.r.t.* S.

In this paper, we use a modified notion of a argumentation semantics *acceptable* given in [9] that captures defeasible provability in Defeasible Logic (DL) [12] with ambiguity blocking. We refer the reader to [13], [14], [15] for a detailed discussion on ambiguity blocking and the difference between ambiguity blocking and ambiguity propagating non-monotonic logics.

Definition 1: A layer-n argument A for p is *acceptable* w.r.t a set of layer-n arguments S if A is finite, and
1. A is a layer-n strict argument, or
2. every layer-n argument attacking A is undercut by S.

Based on this concept we proceed to define justified arguments and justified literals.

Definition 2: Let $T = (R, N)$ be an LAS theory. We define J_i^n as follows:
(1) $JArgs^0 = \emptyset$;
(2) $SArgs_n = JArgs^{n-1}$;
(3) $J_0^n = \emptyset$;
(4) $J_{i+1}^n = \{a \in Args_n \cup SArgs_n \mid a$ is acceptable w.r.t. $J_i^n\}$;
(5) $JArgs^n = \cup_{i=1}^{\infty} J_i^n$ is the set of justified layer-n arguments of an LAS theory T.
We can now give the definition of a finite set of literals denoting the set of layer-n conclusions C_n of T as the set of the conclusions of the arguments in $JArgs^n$. A literal p is layer-n justified if it is the conclusion of an argument in $JArgs^n$. Since arguments cannot attack stronger arguments, conclusions made in lower layers are subsumed to upper layers.

Proposition 1: (Conclusion Subsumption) Let $T = (R, N)$ be an LAS theory. Then, the following holds for $1 \leq n \leq N$

$$C_{n-1} \subseteq C_n$$

The process of building justified arguments is incremental, that is, first we build all layer-1 arguments, and then layer-2, so on towards layer-N. Figure 2 illustrates the process of generating justified arguments incrementally. Thus, when we

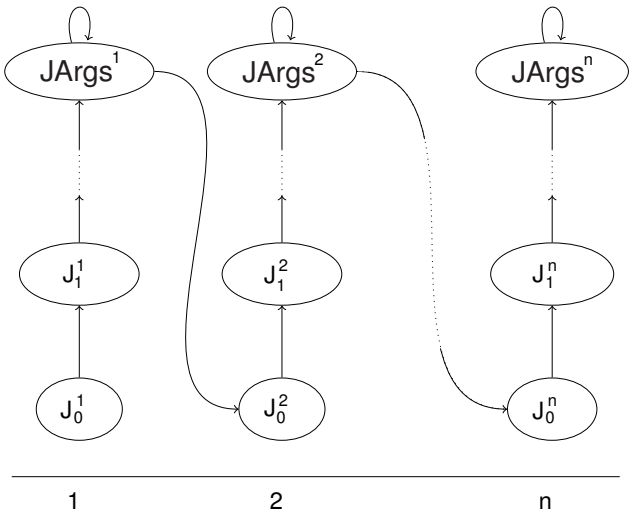


Fig. 2. Incremental generation of justified arguments starting from layer-1.

develop an implementation of LAS, it is convenient to have a notion referring to the argumentation of each individual layer.

Definition 3: Let $T = (R, N)$ be an LAS theory. Let $JArgs^n$ be the set of justified layer-n arguments of T. The layer-n argumentation of T is defined inductively as follows:
1. $T_0 = (\emptyset, \emptyset)$.
2. $T_1 = (\emptyset, \mathbb{R}_1)$ is layer-1 argumentation where $\mathbb{R}_1 = R_1$;
3. $T_n = (JArgs^{n-1}, \mathbb{R}_n)$ is layer-n argumentation where $\mathbb{R}_n = R_1 \cup ... \cup R_n$;

We should note that, by the definition, layer-n subsumes (includes) layer-$(n-1)$ rules. That is, unlike other layered or hierarchical approaches (e.g., [6], [16], [10]) lower layer rules (more confident rules) are reused in higher layers. This feature is important because facts and rules with different degrees of confidence can interact similarly to Possibilistic Logic approaches. [3], [11] and Fuzzy-logic approaches. We now define and discuss a set of fuzzy-argumentation operators corresponding to the Fuzzy-logic operators in the next section.

V. Fuzzy-Argumentation Operators of LAS

The reasoning method of LAS is somewhat similar to Possibilistic Logic approaches [3], [11] such that facts and rules with different degrees of confidence can be combined as shown in Example 2. For instance, let q and p be literals, $LI(A)$ the layer index of an argument A, and $Lc(Q)$ the layer index of a propositional formula Q in an LAS theory T, then

it is easy to see that the following operators corresponding to the Fuzzy-logic operators [17, p.142] can be defined for LAS:

1) $Lc(q) = min(\{\infty, LI(A) \mid \text{justified argument } A \text{ for } q \text{ in } T\})$
2) $Lc(q \wedge p) = max(Lc(q), Lc(p))$ which is the layer index of concluding q and p given T.
3) $Lc(q \vee p) = min(Lc(q), Lc(p))$ which is the layer index of concluding q or p given T.

If the layer index of a propositional formula Q is n in T, the degree of confidence of Q in T is the same as the layer-n in T. With these operators, an agent can evaluate the layer index of any classical logic formula. We should note that we can even evaluate formulas containing logical-OR ($\vee$) operator, which is not allowed in LAS. For instance, in Example 2, the layer index that c or $\neg v$ are true can be obtained as follows:

$$\begin{aligned} Lc(c \vee \neg v) &= min\{LI(B), LI(D)\} \\ &= min\{2, 1\} = 1 \end{aligned}$$

where B is a justified argument for c and D is a justified argument for $\neg v$.

However, unlike Fuzzy Logic approaches, the value of $Lc(\sim q)$ is infinite (∞) if there is a justified argument for q, because the agent has already committed to believe the conclusion of q until there is a contrary evidence. Thus, for any literal q, $Lc(q \wedge \sim q) = \infty$ meaning that it is not possible to conclude both q and $\sim q$. In Fuzzy-logic, the fuzzy truth value of $\sim q$ is considered as the mirror image of q [17, p.136]: $F(q \wedge \sim q) = min(F(q), 1 - F(q))$, which is not reasonable because if $F(q) = 0.5$, then $F(\sim q) = 0.5$ meaning q is both true and false.

In addition, unlike Fuzzy Logic approaches, LAS does not rely on measurements on the degree of belief for each rules and evidence. The layers represent *relative precedence* of rules and relative risk/benefits on conclusions. For example, if an association rule r_1 has been observed more frequently than r_2, an agent might simply prefer r_1 over r_2 whereas a possibilistic logic requires certainty degrees for both rules and facts in order to draw conclusions. Thus, LAS can be more suitable when only relative preference over rules and evidence can be obtained. In modelling certain domains, LAS also can has significant computational advantage in comparison to Fuzzy-logic approaches, since LAS operates in an interval of natural numbers $[1, N]$ in comparison to an interval of real-numbers $[0, 1]$ in Fuzzy-logic approaches.

VI. Designing Fuzzy Controllers with LAS

Let us now compare LAS with Fuzzy-logic for designing a fuzzy controller. The temperature controller illustrated in Figure 1 can be represented in an LAS theory $T = (\{R_1\}, 2)$ consisting of two layers and the following two rules in layer-1:

$$R_1 = \{cold \rightarrow high, \quad hot \rightarrow low\}$$

To make our case more interesting, suppose the water temperature is $50°C$ and the fuzzifier[2] returns that $50°C$ is certainly

[2]Referring to the process of mapping each crisp input value (e.g., temperature) to the degree of membership to each class (e.g., cold, hot).

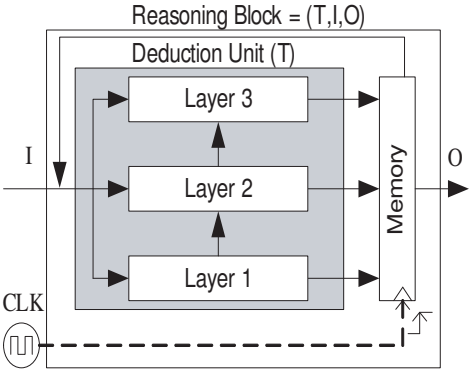


Fig. 3. A reasoning block with three layers.

cold (layer-1) and it just might be *hot* (layer-2). Thus, we add the following rules:

$$R_1 = \{\rightarrow cold\} \qquad R_2 = \{\rightarrow hot\}$$

Then, we can conclude that $high$ is layer-1 justified and low is layer-2 justified. Similarly to Fuzzy-logic, there can be many methods for calculating the *crisp* output from these conclusions. We can use similar aggregation and defuzzyfication functions, but we can just take the most confident conclusion: $high$. Then, we can set the voltage to a pre-assigned value for $high$ (e.g., 20 volt). This is a reasonable choice for rational agents: given the evidence, the agent came to a decision (the best possible commitment) to set the voltage to $high$ no matter how uncertain the conclusion was. Alternatively, we can take weighted (by layer-index) average of the conclusions.

Unlike fuzzy logic, inference rules can be fuzzy too. For example, if the agent is not so certain about rule $cold \rightarrow high$, the agent can put it into layer-2: R_2. Then, both $high$ and low are layer-2 justified. In this case, the agent can choose either of the conclusions to set the output voltage.

VII. Using LAS for Controller Chip Design and Implementation

LAS has been used to implement agent chips. An agent chip is a hardware implementation of a high-level agent specification. It is a digital circuit that produces outputs in reaction to its inputs that are connected to the agent's environment.

The basic building block of an agent chip is the *reasoning block* shown in Figure 3. An agent chip can have one or more reasoning blocks. A reasoning block of an agent chip is specified by a knowledge base T, an input specification I, and an output specification O. The deduction unit of a reasoning block is implemented on FPGAs or ASICs as *combinational logic circuits* , which is organized in layers corresponding to the layers of the knowledge base T.

Figure 4 shows an example agent architecture called *Agent Block* which consists of three reasoning blocks: a belief generator (B), a goal generator (G), and an action (or plan) generator (A). Each agent block is an autonomous system that performs actions in order to achieve a certain desired state D. The reasoning blocks derive conclusions from their input and

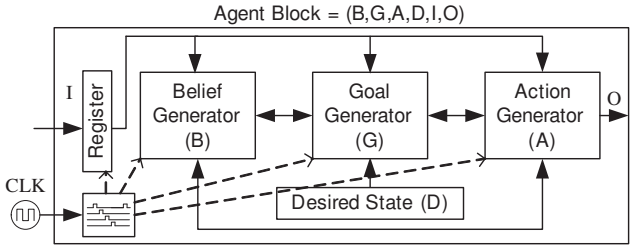


Fig. 4. Agent Block architecture: an example agent architecture consisting of three control blocks: belief generator (B), goal generator (G), and action generator (A)

feed the conclusions to other reasoning blocks or registers. The reasoning blocks are driven by the clock signal (CLK) shown in the figure. Consequently, the frequency of the clock is determined by the time required to draw all of the conclusions in each deduction units. That is, the computation bottle-neck is in the reasoning blocks because they are responsible for drawing all conclusions.

However, by using LAS, the reasoning blocks can be implemented as pure combinational logic circuits. Computational characteristics of combinational logic circuits are highly parallel such that each reasoning block can compute all of conclusions almost instantaneously. For instance, our implementation of an agent chip consisting of 16,000 rules on a *Xilinx*^{TM} Spartan-3 FPGA chip (currently each chip costs less than US$9) can operates in 140MHz clock frequency (i.e., less than 8×10^{-9} seconds for each cycle) whereas conventional approaches based on more expensive CPUs (e.g., 3Ghz Pentium 4 CPU) require few minutes for each cycle.

VIII. Discussion

In this paper we defined an argumentation system extended with subsumption concept and varying degrees of confidence along with interesting fuzzy-argumentation operators. LAS can provide conceptual decomposition as well as behavioral decomposition of agent systems through rule and conclusion subsumption. The conclusions of LAS theories contain the confidence degree information so that we can build agents to better cope with dynamic situations by adjusting the acceptance degree of confidence depending on the risks involved for each situation.

Fuzzy-argumentation operators of LAS are independent of the definitions of *acceptable*. That is, the same definition of fuzzy-argumentation operators can be used with different argumentation semantics. Thus, system specifications containing fuzzy-argumentation operators are also independent of argumentation semantics of LAS. For instance, the example specification of the simple temperature controller shown in Section VI is independent of argumentation semantics of LAS.

Each layer of LAS can also have a superiority relation for rules (and subsumed rules) in the layer similarly to standard Defeasible Logic as a logical theory with a superiority relation can be transformed into an equivalent logical theory without a superiority relation [18].

IX. Conclusion

Similarly to Possibilistic Logic approaches [3], [11], facts and rules with different degrees of confidence can be combined as shown in Example 2. However, unlike Fuzzy Logic approaches, fuzzy-logic like fuzzy operators can be defined in LAS without paradoxes. We defined fuzzy-argumentation operators and showed that how they can be used to specify controllers using the notion of fuzziness. Another difference of LAS from Fuzzy Logic approaches is that LAS does not rely on measurements on the degree of belief for each rules and facts because the layers represent *relative precedence* of rules and relative risk/benefits on conclusions.

References

[1] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni, "An abstract, argumentation-theoretic approach to default reasoning," *Artificial Intelligence*, vol. 93, pp. 63–101, 1997.

[2] H. Prakken and G. Sartor, "Argument-based extended logic programming with defeasible priorities." *Journal of Applied Non-Classical Logics*, vol. 7, no. 1, 1997.

[3] D. Dubois, J. Lang, and H. Prade, "Possibilistic logic," in *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, D. Gabbay, C. J. Hogger, and J. A. Robinson, Eds. Oxford: Oxford University Press, 1994, pp. 439–513.

[4] P. Besnard and A. Hunter, "Practical first-order argumentation," in *AAAI'2005*. MIT Press, 2005, pp. 590–595.

[5] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[6] E. Amir and P. Maynard-Zhang, "Logic-based subsumption architecture," *Artif. Intell.*, vol. 153, no. 1-2, pp. 167–237, 2004.

[7] I. Song and G. Governatori, "Designing agent chips," in *5th International Conference on Autonomous Agents and Multi-Agent Systems*, P. Stone and G. Weiss, Eds. ACM Press, 10–12 May 2006, pp. 1311–1313.

[8] I. Song and G. Goverantori, "A compact argumentation system for agent system specification," in *Proceedings of the Third Starting AI Researchers' Symposium: STAIRS'06*, vol. 142, 2006.

[9] G. Governatori, M. J. Maher, G. Antoniou, and D. Billington, "Argumentation semantics for defeasible logics," *Journal of Logic and Computation*, vol. 14, no. 5, pp. 675–702, 2004.

[10] K. Konolige, "Hierarchic autoepistemic theories for nonmonotonic reasoning," in *Non-Monotonic Reasoning: 2nd International Workshop*, 1989, vol. 346, pp. 42–59.

[11] C. I. Chesnevar, G. R. Simari, T. Alsinet, and L. Godo, "A logic programming framework for possibilistic argumentation with vague knowledge," in *Proc. AUAI '04: the 20th conference on Uncertainty in artificial intelligence*, 2004, pp. 76–84.

[12] D. Nute, "Defeasible logic," in *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford: Oxford University Press, 1994, pp. 353–395.

[13] L. A. Stein, "Resolving ambiguity in nonmonotonic inheritance hierarchies," *Artif. Intell.*, vol. 55, no. 2-3, pp. 259–310, 1992.

[14] D. Touretzky, J. Horty, and R. Thomason., "A clash of intuitions: The current state of nonmonotonic multiple inheritance systems," in *Proc. IJCAI-87*. Morgan Kaufmann, 1987, pp. 476–482.

[15] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher, "A flexible framework for defeasible logics," in *AAAI 2000*, 2000, pp. 405–410.

[16] M. Wooldridge, P. McBurney, and S. Parsons, "On the meta-logic of arguments," in *Proc. AAMAS '05*, 2005, pp. 560–567.

[17] H. Zimmermann, *Fuzzy Set Theory — and Its Applications*, 2nd ed. Kluwer Academic Publishers, 1991.

[18] M. J. Maher, "Propositional defeasible logic has linear complexity," *Theory and Practice of Logic Programming*, vol. 1, no. 6, pp. 691–711, 2001.