A Compact Argumentation System for Agent System Specification

Insu Song and Guido Governatori

School of Information Technology & Electrical Engineering
The University of Queensland, Brisbane, QLD, 4072, Australia
e-mail: {insu,guido}@itee.uq.edu.au

Abstract. We present a non-monotonic logic tailored for specifying compact autonomous agent systems. The language is a consistent instantiation of a logic based argumentation system extended with Brooks' subsumption concept and varying degree of belief. Particulary, we present a practical implementation of the language by developing a meta-encoding method that translates logical specifications into compact general logic programs. The language allows n-ary predicate literals with the usual first-order term definitions. We show that the space complexity of the resulting general logic program is linear to the size of the original theory.

Keywords. Argumentation, Automated Reasoning, Agent

1. Introduction

Over the past years, we have witnessed massive production of small electronic consumer devices such as cell phones, set-top boxes, home network devices, and MP3 players. The size of the devices gets smaller and the behaviors of the devices get more and more complex. In order to survive in the current competitive market, vendors now must scramble to offer more variety of innovative products faster than ever before because most of modern consumer electronic devices have comparatively low run rates and/or short market windows [7].

One solution for reducing the development cost and time is developing a more expressive and intuitive specification language for describing the behaviors of products. But, we must make the resulting systems compact and efficient to meet the current market demands: smaller systems and longer battery lifespan. One promising candidate language is a nonmonotonic logic-based agent architecture because nonmonotonic logics are close to our natural languages and many agent models are suitable for specifying complex autonomous behaviors. In short, our aim is to develop a more expressive logic-based language than existing industrial specification languages, such as Ladder Logic in PLCs (Programmable Logic Controllers), while maintaining its simplicity, robustness (when implemented), and implementability on low-profile devices.

However, existing logical approaches [1,21,10,4,19,8,6,3] are not suitable for this purpose, because they suffer from the following major shortcomings to be embedded in small low powered devices: (a) they have difficulties in expressing behaviors, (b) they

require high computing power, and (c) they are not suitable for mission critical applications as they require sophisticated theorem provers running on a high powered CPU. We solve these problems (1) by devising a nonmonotonic logic that can be mapped into a computationally compact structure that is suitable for hardware implementation; (2) by allowing expressions of relative certainties in knowledge bases; (3) by allowing decomposition of systems into parallel interacting behaviors similarly to the subsumption architecture [5].

In particular, we develop a layered argumentation system called *LAS* that extends a logic based proposal of argumentation with subsumption concept and varying degree of confidence. The reasoning mechanism of each layer is the argumentation system and more confident layers are subsumed by less confident layers. Moreover, we present a practical implementation of LAS by developing a meta-encoding method that translates LAS into general logic program. Unlike other existing implementations of argumentation systems, the language allows n-ary predicate literals with the usual first-order term definitions including function expressions. Importantly, we show that the size of the resulting general logic program is linear to the size of the original theory. Similar meta-encoding schemas can be developed for other variants of logic based argumentation systems under our framework. Thus, we believe the framework will also provide a platform for extending and developing practical implementations of logic based argumentation systems.

In this paper we detail the logical language (LAS), the mapping of the language into general logic program, and benefits it offers in comparison with other approaches. The paper is structured as follows. In the next section we discuss a behavior based decomposition and its relation with LAS. In Section 3, we define the Layered Argumentation System (LAS) and the formal semantics of the underlying language. After that, in Section 5, we present a meta-encoding schema which transforms first-order knowledge bases of LAS into general logic programs. Then, in Section 6 we compare LAS with other logic base approaches and conclude with some remarks in Section 7.

2. Behavior Based Decomposition

In [5], Rodney Brooks introduced a subsumption that decomposes a system into several layers of (possibly prioritized) parallel behaviors of increasing levels 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 [1]. To overcome the limitation, we need to develop a logic based subsumption architecture (e.g., [1]). To do this, we need to introduce a varying degree of confidence and a subsumption architecture into a logical system.

For instance, let us consider a cleaning robot. A typical functional decomposition of this system might resemble the sequence:

sensors→perception→modelling→planning→task selection→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 levels 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 urgent behaviors. However, this is not always the case because a strong will can suppress a reactive behavior. Therefore, since the layers of LAS represent the levels of confidence instead of competence, the layers of the subsumption architecture [5] do not exactly correspond to the layers of LAS as shown below.
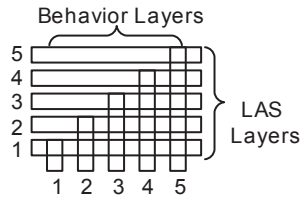
In this figure, the bottom layer is the most confident layer of LAS whereas the right most layer is the most competent behavior layer. It shows that less competent behaviors tend to be related with more confident layers whereas more competent behaviors tend to spread over several layers of LAS. As we will see in the following sections, the subsumption concept of [5] is used in decomposing concepts as well because a less confident knowledge base subsumes more confident knowledge bases by rule subsumption in LAS.

Example 1 *Let us consider an example specification for the part that controls a vacuum cleaning unit of the cleaning robot. Suppose that the robot performs vacuuming action (v) if it detects (on sensor sA) that the area is dirty (d), but it stops the action if it detects some water (w) on sensor sB. That is, we have two parallel behaviors: avoiding water and cleaning. However, avoiding water has higher priority than cleaning. This specification can be represented as a set of defeasible rules decomposed into two levels of rules as follows:*

$$R_1 = \{r_1 : sA \rightarrow d, \ r_2 : sB \rightarrow w, \ r_3 : w \rightarrow \neg c\}$$

$$R_2 = \{r_4 : d \rightarrow c, \ r_5 : c \rightarrow v\}$$

The arrows represent defeasible inferences. For instance, $sA \rightarrow d$ is read as 'if sA is true, then it is usually dirty'. The levels represent relative confidences between levels such that level-n conclusions are more confident than level-$(n+1)$ conclusions. Then, if an area is both dirty and wet, the vacuuming unit will be turned off: v is not true. The reason is that since w is a level-1 conclusion by r_2, $\neg c$ is a level-1 conclusion by r_3. As level-1 conclusions are more confident than level-2 conclusions, $\neg c$ is also a level-2 conclusion overriding c in R_2. Thus, we cannot conclude v.

3. Layered Argumentation System

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. From now on, let *level-n* denote the degree of confidence of layer-*n*.

An LAS theory is a structure $T = (R, N)$ where $R = \{R_1, ..., R_n, ..., R_N\}$ is a set of finite sets of rules where R_n is a set of level-*n* rules and N is the number of layers. We now define layers of an LAS theory and their conclusions.

Definition 1 *Let $T = (R, N)$ be an LAS theory. Let C_n be a finite set of literals denoting the set of layer-n conclusions of T. Let n be a positive integer over the range of $[1, N]$. The layers of T are defined inductively as follows:*

1. $T_0 = (\emptyset, \emptyset)$.
2. $T_1 = (\emptyset, \mathbb{R}_1)$ is layer-1 theory where $\mathbb{R}_1 = R_1$;
3. $T_n = (C_{n-1}, \mathbb{R}_n)$ is layer-*n* theory 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., [1,21,14]) lower layer rules (more confident rules) are reused in higher layers. This feature is important because facts and rules with different levels of belief can interact similarly to Possibilistic Logic approaches [8,6] and Fuzzy Logic approaches. For example, the confidence of the resulting argument formed by a set A of rules is the same as the confidence of the rule having the *minimum* confidence in A. We will see an example of this (Example 2) after we define the semantics of LAS. We will also define and discuss a set of operators corresponding to the Fuzzy Logic operators in Section 6.2.

As for the semantics of the language, we modify the argumentation framework given in [11] 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:
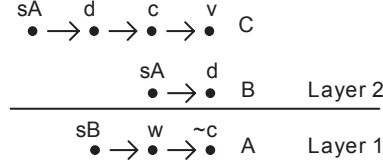
1. If $b_1, ..., b_n$ label the children of h then there is a rule in R with body $b_1, ..., b_n$ and head h.
2. The arcs in a proof tree are labelled by the rules used to obtain them.

Given a layer T_n of an LAS theory T, the set of layer-*n* arguments is denoted as $Args_{T_n}$ which also denotes the set of arguments that can be generated from $\mathbb{R}_n$. The degree of confidence of an argument in $Args_{T_n}$ is level-*n*. Thus, a layer-*n* argument of T is more confident than layer-$(n+1)$ arguments of T. We define $Args_{T_0}$ to be the empty set. 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.

We now introduce a set of usual notions for argumentation system. An argument A *attacks* an argument B if the conclusion of A is the complement of a conclusion of B and the confidence of B is equal to or less than A. A set S of arguments attacks an argument

B if there is an argument *A* in *S* that attacks *B*. An argument *A* is *supported* by a set of arguments *S* if every proper subargument of *A* is in *S*. An argument *A* is *undercut* by a set of arguments *S* if *S* supports an argument *B* attacking a proper subargument of *A*.

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

$$\begin{array}{ccccccc}
sA & d & c & v \\
\bullet \rightarrow & \bullet \rightarrow & \bullet \rightarrow & \bullet & C \\
\end{array}$$

$$\begin{array}{ccc}
sA & d \\
\bullet \rightarrow & \bullet & B \qquad \text{Layer 2} \\
\end{array}$$

$$\begin{array}{ccc}
sB & w & \sim c \\
\bullet \rightarrow & \bullet \rightarrow & \bullet & A \qquad \text{Layer 1} \\
\end{array}$$

A is a layer-1 argument for $\sim c$, *and thus it is also a layer-2 argument because layer-1 arguments are more confident than layer-2 arguments. B is a layer-2 argument for d and a sub-argument of C. We should note that we have B because level-1 rule* r_1 *is subsumed by layer-2. That is, level-2 evidence sA and level-1 rule* r_1 *are combined to produce a level-2 argument. In addition, since there is no level-1 evidence for sA, there is no layer-1 argument for d. This cannot be represented in exiting layered approaches such as [14]. Unlike Fuzzy Logic approaches [8,6], the inference process is very simple and more intuitive, since LAS does not require number crunching nor require subjective measures of confidence.*

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 an argument *A* and a set *S* of 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*.

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* $J_i^{T_n}$ *as follows:*

- $JArgs^{T_0} = \emptyset$;
- $J_0^{T_n} = JArgs^{T_{n-1}}$;
- $J_{i+1}^{T_n} = \{a \in Args_{T_n} \mid a$ is acceptable w.r.t. $J_i^{T_n}\}$;
- $JArgs^{T_n} = \cup_{i=1}^{\infty} J_i^{T_n}$ is the set of justified layer-*n* arguments.

We can now give the definition of C_n as the set of conclusions of the arguments in $JArgs^{T_n}$. A literal *p* is level-*n* justified (denoted as $T_n \vDash_l p$) if it is the conclusion of an argument in $JArgs^{T_n}$. In Example 2, literal $\sim c$ is both level-1 and level-2 justified, but literal *v* is not justified because argument *C* is undercut by *A*.

We now give two definitions of *acceptable* given in [11]. The following is an argumentation semantics that corresponds to Dung's skeptical semantics (called grounded semantics) [9,10] which has been widely used to characterize several defeasible reasoning systems [10,4,19].

Definition 3 *An argument A for p is* acceptable *w.r.t a set of arguments S if A is finite, and every argument attacking A is attacked by S.*

The following definition is a modified notion of acceptable in order to capture defeasible provability in Defeasible Logic (DL) [17] with ambiguity blocking [11].

Definition 4 *An argument A for p is* acceptable *w.r.t a set of arguments S if A is finite, and every argument attacking A is undercut by S.*

In this paper, we use this ambiguity blocking argumentation semantics as the semantics of LAS. But, the grounded semantics can also be easily adopted to LAS.

4. An Implementation of LAS

We obtain the meta-program representation of an LAS theory from the meta-program formalization of Defeasible Logic (DL) given in [16] by removing defeaters, strict rules, priority relations, and converting the relationship between strict rules and defeasible rules in DL in to the relationship between layer-$(n-1)$ and layer-n in LAS. Thus, it is also an ambiguity blocking Dung-like argumentation system [11]. The details of the meta-program representation of an LAS theory $T = (R, N)$ and its layers are now described. First, we obtain the *conclusion-meta-program* $\Pi_C(T)$ consisting of the following general logic programs of each layer-n where $1 \leq n \leq N$:

C1. $\text{conclusion}_n(x) \colon\text{-} \text{conclusion}_{n-1}(x)$.
C2. $\text{conclusion}_n(x) \colon\text{-} \text{supported}_n(x), \text{not supported}(\sim x), \text{not conclusion}_{n-1}(\sim x)$.

where not denotes the negation as failure and $\sim x$ maps a literal x to its complement. Next, we obtain the *rule-meta-program* $\Pi_R(T)$ consisting of the following general logic programs for each rule $(q_1, ..., q_m \rightarrow p) \in \mathbb{R}_n$ of each layer-n:

R1. $\text{supported}_n(p) \colon\text{-} \text{conclusion}_n(q_1), ..., \text{conclusion}_n(q_m)$.

Then, the corresponding general logic program of T is

$$\Pi(T) = \Pi_C(T) \cup \Pi_R(T).$$

In [11], it is shown that the following theorem holds for the ambiguity blocking argumentation system.

Theorem 1 *Let D be a defeasible theory, p be a literal. Then, $D \vdash +\partial p$ iff p is justified, where $+\partial p$ means p is defeasibly provable.*

From this theorem and the correctness of the meta-program of Defeasible Logic [16], We obtain the following theorem. Let $\vDash_\kappa$ denote logical consequence under Kunen's semantics of logic programs [15].

Theorem 2 *Let T_n be layer-n of an LAS theory T. Let D_n be the meta-program counter part of T_n.*

  1. $D_n \vDash_\kappa \text{conclusion}(p)$ iff p is level-n justified (i.e., $T_n \vDash_l p$).
  2. $D_n \vDash_\kappa \neg\text{conclusion}(p)$ iff p is not level-n justified (i.e., $T_n \nvDash_l p$).

The following theorem is a direct consequence of rule C1.

Theorem 3 *(Conclusion subsumption) Let T_n be layer-n of LAS theory T and T_{n+1} be layer-$(n+1)$ of T. Let C_n be the set of conclusions of T_n and C_{n+1} be the set of conclusions of T_{n+1}. Then $C_n \subseteq C_{n+1}$.*

The following *layer-consistency* is the result of [16] (correctness of the meta-program of defeasible logic) and consistency of defeasible logic.

Theorem 4 *(Layer consistency) Let T_n be layer-n of an LAS theory T. Then, for each literal p, if $T_n \models_l p$ then $T_n \nvDash_l \sim p$.*

Then, from theorem 3 and theorem 4, we can show that T is consistent.

Theorem 5 *(LAS consistency) Let T be an LAS theory. Let $T \vDash q_n$ denote that $T_n \models_l q$. Then, for all $1 \leq n \leq N$, if $T \vDash q_n$ then $T \nvDash (\sim q)_m$ for all $1 \leq m \leq N$.*

As $\Pi(T)$ is simply the union of the meta-program counterpart D_n of each layers of T and it is formed through a level mapping of each D_n to form a hierarchical program such that no literals appearing layer-n program appear in layer-$(n-1)$ rules, the following theorem holds.

Theorem 6 *Let $T = (R,N)$ be an LAS theory and $\Pi(T)$ be the meta-program counterpart. Then, the following holds:*
$T \vDash q_n$ *iff* $\Pi(T) \vDash_\kappa$ conclusion$_n(q)$.
$T \nvDash q_n$ *iff* $\Pi(T) \vDash_\kappa \neg$conclusion$_n(q)$.

5. Meta-Encoding

We now formally define the meta-encoding schema that translates LAS theories into propositional general logic programs. First, we define two literal encoding functions that encode literals in an LAS theory to previously unused positive literals. Let q be a literal and n a positive integer. Then, these functions are defined below:

$$Supp(q,n) = \begin{cases} p_n^{+s} & \text{if } q \text{ is a positive literal } p. \\ p_n^{-s} & \text{if } q \text{ is a negative literal } \neg p. \end{cases}$$

$$Con(q,n) = \begin{cases} p_n^{+} & \text{if } q \text{ is a positive literal } p. \\ p_n^{-} & \text{if } q \text{ is a negative literal } \neg p. \end{cases}$$

$Supp(q,n)$ denotes a support of q at layer-n and $Con(q,n)$ denotes a conclusion q at layer-n. $Supp(q,n)$ corresponds to supported$_n$(q). $Con(q,n)$ corresponds to conclusion$_n$(q). Let $ConA(A,n)$ be a set of new positive literals obtained from a set A of literals by replacing each literal $q \in A$ by $Con(q,n)$: $ConA(A,n) = \{Con(q,n)|q \in A\}$. With these functions, we now define the meta-encoding schema.

Let $T = (R,N)$ be an LAS theory, $\Pi(T)$ the corresponding meta-program, and L the set of all propositional letters in T. Then $H_T = L \cup \sim L$ is the Herbrand universe of $\Pi(T)$ where $\sim L = \{\sim p|p \in L\}$. The translated Herbrand base $G(T)$ of $\Pi(T)$ is obtained according to the following guidelines for each layer-n ($1 \leq n \leq N$):

G1: For each $q \in H_T$, add

$$Con(q,n) \leftarrow Con(q,n-1).$$

G2: For each $q \in H_T$, add

$$Con(q,n) \leftarrow Supp(q,n), not\ Supp(\sim q,n), not\ Con(\sim q,n-1).$$

G3: For each $r \in \mathbb{R}_n$, add

$$Supp(C(r),n) \leftarrow ConA(A(r),n).$$

For most of LAS theories, this direct translation of T results in a lot of redundant rules that will be never used for generating conclusions. But, if we know the set of all the literals that will ever be supported, we can reduce the number rules in G2 by replacing 'For each $q \in H_T$' by 'For each $q \in SL_n$' where SL_n is the set of all supportive literals in layer-n of T. SL_n can be obtained as follows:

$$SL_n = \{C(r)|r \in \mathbb{R}_n\}$$

Let $G2(T)$ be the set of rules introduced by G2 in $G(T)$. Then, we can also reduce the number rules in G1 by replacing 'For each $q \in H_T$' by 'For each $q \in CL_n$' where CL_n is the set of all conclusion literals in layer-n of T. This can be obtained as follows:

$$CL_0 = \emptyset$$
$$CL_n = CL_{n-1} \cup \{p|Con(p,n-1) \in G2(T)\}$$

For example, let's consider an LAS theory $T = (\{R_1\},1)$ where $R_1 = \{\rightarrow sA\}$. The corresponding meta-program $G(T)$ of T is (after removing literals with $n=0$):
$sA_{1.}^{+s}.$
$sA_1^+ :\text{-} sA_1^{+s}, not\ sA_1^{-s}.$

Furthermore, even if we extend the language of LAS to allow for n-ary predicate literals that can be formed by the usual inductive definitions for classical logic, this meta-encoding can be used to convert first-order LAS theories. For example, if we replace sA with $sA(\mathsf{x})$ where x is a variable, then the corresponding logic program becomes:

$sA_1^{+s}(\mathsf{x}).$
$sA_1^+(\mathsf{x}) :\text{-} sA_1^{+s}(\mathsf{x}), \mathsf{not}\ sA_1^{-s}(\mathsf{x}).$

Let us consider an LAS theory $T = (R,N)$ containing x unique rules in each layer. Then, the total number of rules is $X = xN$. The number of rules and facts created by the guidelines is bounded by the following equation:

$$|G(T)| \leq (X)(0.5 + 1.5N)$$

That is, the size of $G(T)$ is linear to the size of T. In fact, if $N = 1$, $|G(T)| \leq 2|T|$. In practice many subsumed rules can be removed because not all of the rules in a layer interact with other layers. For instance, in Example 1, if we don't need level-2 conclusions of sB, w and $\sim c$, there is no need to subsume r_2 and r_3 in layer-2.

6. Comparisons With Other Approaches

6.1. Hierarchical Approaches

Our approach in decomposing systems differs from meta-hierarchical approaches [21] and functional decompositions of knowledge-bases [13]. Unlike hierarchical approaches, the layers in LAS represent varying degree of belief such as 'Jane might be tall', 'Jane is surely tall'. That is, there are no layers representing beliefs about own beliefs and so on. Unlike functional decompositions, subsumption architecture decomposes a system based on the confidence degree of knowledge: one layer is more or less confident than the other layers rather than more or less complex (or dependant). It also differs from preference based logics which use preferences only to resolve conflicts. A knowledge base of a preference based logic (e.g., Defeasible Logic [16,2]) basically corresponds to a layer in LAS. The reason for this is because priority relations between rules (and defeaters in Defeasible Logic) can be represented as just defeasible rules.

The layered structure of LAS is similar to the hierarchic autoepistemic logic (HAEL) [14] in which conclusions of lower layers are stronger than higher layers. However, HAEL has no notion of support (evidence) used in LAS. Thus, it has some limitations such as inconsistent extensions. In LAS, the notion of support is used to prevent credulous conclusions. In addition, HAEL does not allow higher layers to subsume rules of lower layers. That is, in HAEL a level-2 observation (support) can not be used with level-1 rules to produce level-2 conclusions. Most importantly, LAS has a computationally realizable implementation, and the language of LAS is also much more intuitive than autoepistemic logic as formulas are free of modal operators. Logic based subsumption proposed by [1] is also similar to our work, but the language is based on Circumscription, thus it requires second-order theorem provers, and rules are not subsumed.

6.2. Fuzzy Logic Approaches

Similarly to Possibilistic Logic approaches [8,6], facts and rules with different levels of confidence can be combined as shown in Example 2. For instance, let q and p be justified arguments and $L(q)$ be the level of confidence of the conclusion of a justified argument q, then it is easy to see that the following relations corresponding to the Fuzzy Logic operators hold in LAS:

$L(q) = max(\{\text{the levels of the rules used in } q\})$
$L(q \text{ and } p) = max(L(q), L(p))$
$L(q \text{ or } p) = min(L(q), L(p))$

For instance, in Example 2, $L(B) = 2$ because the levels of sA and r_1 are 2 and 1, respectively. However, unlike Fuzzy Logic the value of $L(\text{not } q)$ is undefined or infinite meaning almost impossible because the agent has already committed to believe the conclusion of q until there is a contrary evidence. Thus, $L(q \text{ and not } q)$ is not possible.

However, unlike Fuzzy Logic approaches, LAS does not rely on measurements on the degree of belief for each rules and facts. All the rules in LAS theories can be considered to represent near 100% conditional probability (or agents' commitment despite of possible risks) of the consequents when the corresponding premises are all provable. The layers represent *relative precedence* of rules and relative risk/benefits on conclusions.

For example, in LAS if an association rule r_1 occurs more frequently than r_2, we simply place r_1 to level-1 and r_2 to level-2 whereas a possibilistic logic requires exact figures for both rules and facts, such as certainty degree, in order to draw conclusions. Thus, LAS is more suitable when only relative preference over rules and evidence can be obtained.

6.3. Nonmonotonic Logics

Unlike many existing implementations of argumentation systems, LAS is a concrete implementation with first-order knowledge-bases that considers arguments for and against grounded n-ary predicate literals. [3] proposes an argumentation system that considers arguments for first-order formulas with notion of argument strengths, but it is not clear how a practical reasoning system could be built for it and what the complexity of such systems might be.

LAS incorporates the idea of team defeat [2]. For example, let us consider the following abstract LAS theory:

$S_1 = \{a_1, a_2, b_1, b_2\}$
$R_1 = \{r_1 : b_1 \rightarrow \sim q, \ r_2 : b_2 \rightarrow \sim q\}$
$R_2 = \{r_3 : a_1 \rightarrow q, \quad r_4 : a_2 \rightarrow q\}$

The argumentation system (*PS Logic*) developed by Prakken and Sartor [18,19] cannot conclude $\sim q$ when it is given a priority relation $\{r_1 > r_3, r_2 > r_4\}$ [12] because an attack on a rule with head q by a rule with head $\sim q$ may be defeated by a different rule with head q, but Defeasible Logic [16] can conclude $\sim q$. It is easy to see that both r_1 and r_2 are justified in LAS.

Another interesting problem for testing semantics of nonmonotonic logics is the reinstatement problem that unjustified arguments are considered as reasons for their conclusions which are conclusions of other justified arguments [12]. As an example, let us suppose that birds usually fly (r_1). But, as we all know, penguins usually do not fly (r_2). Now, imagine that genetically modified penguins usually fly (r_3). Then, argument for 'fly' by r_1 is reinstated in PS Logic [12]. This information can be modelled in the following LAS theory:

$R_1 = \{r_3 : gp \rightarrow f, \ gp \rightarrow p, \ p \rightarrow b\}$
$R_2 = \{r_2 : p \rightarrow \sim f\} \quad R_3 = \{r_1 : b \rightarrow f\}$

We should note that r_1 is not a justified argument in LAS unlike PS Logic and standard Defeasible Logic.

7. Conclusion

This paper presented an argumentation system extended with concepts of subsumption and varying degree of confidence along with its interesting properties. 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 level information so that agents can better cope with dynamic situations by adjusting the acceptance level of confidence depending on the risks involved for each situation.

The meta-program $G(T)$ contains only three types of simple rules that can be easily represented as simple combinational logics. A mapping of a propositional LAS system to

```
module testModule( sAcp1, sBcp1, vcp2);      assign dcp2 = dcp1;
  input sAcp1, sBcp1;                         assign dcp2 = dsp2;
  output vcp2;                                assign sAcp2 = sAcp1;
  wor wsp2, dcp2, dcp1, wsp1, sAcp2, sBcp2;   assign sBcp2 = sBcp1;
  wor ccn2, ccn1, csp2, vcp2, wcp2, dsp2;     assign ccn2 = ccn1 | csn2 & ~csp2;
  wor dsp1, wcp1, csn1, csn2, ccp2, vsp2;     assign vcp2 = vsp2;
  wire sAcp1;                                 assign wsp2 = sBcp2;
  wire sBcp1;                                 assign wcp2 = wcp1 | wsp2;
  assign ccn1 = csn1;                         assign csp2 = dcp2;
  assign dcp1 = dsp1;                         assign dsp2 = sAcp2;
  assign wsp1 = sBcp1;                        assign csn2 = wcp2;
  assign wcp1 = wsp1;                         assign ccp2 = csp2 & ~csn2 & ~ccn1;
  assign dsp1 = sAcp1;                        assign vsp2 = ccp2;
  assign csn1 = wcp1;                        endmodule
```

Figure 1. This figure shows a compilation result of the vacuum controller unit into Verilog HDL code. This controller has two inputs and one output. All the symbols represent Boolean variables ranging over the set $\{0,1\}$. For example, sAcp1 represents the state of the level-1 positive conclusion of sA. Value 1 of the state represents that "it is known that sA is true" and value 0 represents "it is unknown that sA is true."
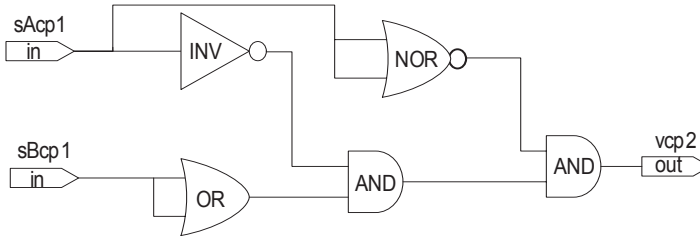
Figure 2. An implementation of the Verilog description of the vacuum control unit in a Xilinx Spartan-3 FPGA. This figure shows an RTL (Register Transistor Logic) level description of the Verilog code for Configurable Logic Blocks (CLBs) of a Spartan-3 FPGA. (Note: the logic circuit does not necessarily represent an optimal logic circuit since it must fit into the CLBs.)

Verilog HDL (Hardware Description Language) has been developed for designing agent silicon chips [20] without a CPU using purely combinational logics and registers. Figure 1 shows an example of the compilation of Example 1 into Verilog HDL. Figure 2 shows an implementation of the Verilog code on a *Xilinx^{TM} Spartan^{TM}*-3 FPGA. Limited forms of first-order LAS theories can also be directly translated into Verilog descriptions using registers, adders, and counters without a CPU making the resulting systems robust and reactive.

To our best knowledge, this is the first practical implementation of a first-order logic based argumentation system with subsumption concept and varying degree of belief.

References

[1] Eyal Amir and Pedrito Maynard-Zhang. Logic-based subsumption architecture. *Artif. Intell.*, 153(1-2):167–237, 2004.

[2] Grigoris Antoniou, David Billington, Guido Governatori, Michael J. Maher, and Andrew Rock. A family of defeasible reasoning logics and its implementation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 459–463, 2000.

[3] Ph Besnard and A Hunter. Practical first-order argumentation. In *AAAI'2005*, pages 590–595. MIT Press, 2005.

[4] Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[5] Rodney A Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[6] Carlos I. Chesnevar, Guillermo R. Simari, Teresa 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*, pages 76–84, 2004.

[7] Suhel Dhanani. FPGAs enabling consumer electronics — a growing trend. *FPGA and Programmable Logic Journal*, June 2005.

[8] Didier Dubois, Jérôme Lang, and Henri Prade. Possibilistic logic. In Dov Gabbay, Christopher J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 439–513. Oxford University Press, Oxford, 1994.

[9] Phan Minh Dung. An argumentation semantics for logic programming with explicit negation. In *ICLP*, pages 616–630, 1993.

[10] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[11] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logics. *Journal of Logic and Computation*, 14(5):675–702, 2004.

[12] John F. Horty. Argument construction and reinstatement in logics for defeasible reasoning. *Artificial Intelligence and Law*, 9(1):1–28, 2001.

[13] J. Huang, N. R. Jennings, and J. Fox. An agent architecture for distributed medical care. In *Intelligent Agents: Theories, Architectures, and Languages*, pages 219–232. Springer-Verlag: Heidelberg, Germany, 1995.

[14] Kurt Konolige. Hierarchic autoepistemic theories for nonmonotonic reasoning. In *Non-Monotonic Reasoning: 2nd International Workshop*, volume 346, pages 42–59. 1989.

[15] Kenneth Kunen. Negation in logic programming. *J. Log. Program.*, 4(4):289–308, 1987.

[16] M. J. Maher and G. Governatori. A semantic decomposition of defeasible logics. In *AAAI '99*, pages 299–305, 1999.

[17] Donald Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 353–395. Oxford University Press, Oxford, 1994.

[18] Henry Prakken and Giovanni Sartor. A system for defeasible argumentation, with defeasible priorities. In *Proc. FAPR '96: the International Conference on Formal and Applied Practical Reasoning*, pages 510–524, London, UK, 1996. Springer-Verlag.

[19] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1), 1997.

[20] Insu Song and Guido Goverantori. Designing agent chips. In *Fifth International Joint Conference on Autonomous Agents & Multi Agent Systems AAMAS06, in print, available at http://eprint.uq.edu.au/archive/00003576/*. ACM Press, 2006.

[21] Michael Wooldridge, Peter McBurney, and Simon Parsons. On the meta-logic of arguments. In *Proc. AAMAS '05*, pages 560–567, 2005.