

Merging Deductive and Abductive Knowledge Bases: An Argumentation Context Approach

Juan Carlos Nieves and Helena Lindgren

Abstract—The consideration of heterogenous knowledge sources for supporting decision making is key to accomplish informed decisions, *e.g.*, about medical diagnosis. Consequently, merging different data from different knowledge bases is a key issue for providing support for decision-making. In this paper, we explore an argumentation context approach, which follows how medical professionals typically reason, in order to merge two basic kinds of reasoning approaches based on logic programs: deductive and abductive inferences. In this setting, we introduce two kinds of argumentation frameworks: deductive argumentation frameworks and abductive argumentation frameworks. For merging these argumentation frameworks, we follow an approach based on argumentation context systems. We illustrate the approach by considering two different declarative specifications of evidence-based medical knowledge into logic programs in order to support informed medical decisions.

Index Terms—Knowledge representation, deductive knowledge bases, abductive knowledge bases.

I. INTRODUCTION

THE knowledge used when reasoning about a medical diagnosis is ideally based on evidence-based medical knowledge generalizable over a large population. However, this knowledge is translated into diagnostic criteria based on consensus among researchers in order to become applicable to a single individual and of practical use in the encounter with a patient. These different types of sources of knowledge make use of different reasoning strategies, which are co-existing and observable in medical professionals' decision making (*e.g.*, causal and diagnostic reasoning) [1]. We acknowledge this, and propose the notion of *default argumentation context system*, meaning that there is at least two supporting perspectives for each claim, where the supplementary part of an argument may be considered being a *meta*-argument, providing strength based on contextual information. An illustrating example is the following: consider the situation where there are diagnostic criteria for a disease, which a patient partly fulfills considering the available observations. However, since the available knowledge is incomplete, a verification is made using an evidence-based medical study where the diagnosis can be supported based on a population study conducted in the area where the patient is living.

Nowadays in distributed systems, the integration of multiple knowledge bases has been taking relevance. Indeed, one can find different approaches in the state of the art (*e.g.* the context-based argumentation framework outlined in [2] and the multi-context systems in [3]). Both approaches aim at utilizing, and bridging different frameworks of interpretation of available observations.

Against this background, we will explore the multi-context systems approach further in this paper with the goal to combine different kinds of argumentation frameworks to generate *informed medical decisions*. To this end, we define two kinds of argumentation frameworks: *deductive and abductive argumentation frameworks*. The interaction between deductive and abductive argumentation frameworks is managed by the so called *bridge rules*. An illustration of this interaction is depicted in Figure 1. Both the deductive and abductive argumentation frameworks are based on logic programs with *negation as failure* and the *well-founded semantics* (WFS) [4]. In particular, we consider WFS for building deductive and abductive arguments. We want to point out that we chose WFS because this semantics is polynomial time computable; moreover, there are logic programming engines which compute WFS, *e.g.*, DLV¹, SMOBELS², XSB³. In order to integrate the deductive and abductive argumentation frameworks, we introduce the so called *default argumentation context systems*. We show that by considering particular argumentation semantics as *the grounded semantics*, one can infer *collective acceptable states* from a default argumentation context system in polynomial time (Proposition 3).

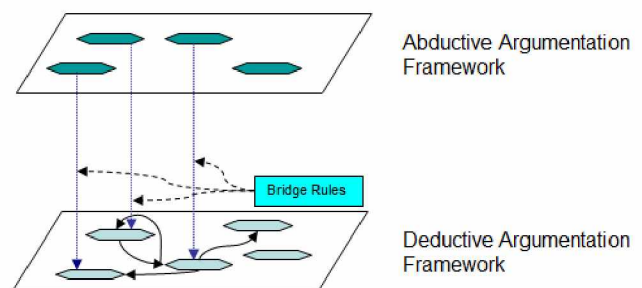


Fig. 1. Combining different knowledge bases by considering different kinds of argumentation frameworks.

Manuscript received on August 1, 2013; accepted for publication on September 30, 2013.

The authors are with the Department of Computing Science, Umeå University, SE-901 87, Umeå, Sweden (e-mail: {jcnieves,helena}@cs.umu.se).

¹<http://www.dbai.tuwien.ac.at/proj/dlv/>

²<http://www.tcs.hut.fi/Software/smodels/>

³<http://xsb.sourceforge.net/>

As a running example, we will use an use-case when a patient shows symptoms that can be evaluated using more than one knowledge source and the diagnosis that is supported by more knowledge sources is preferred. The example involves two physicians, one is a novice and less experienced with the type of disease in focus for diagnosis, and one is an experienced physician. These two typically use different reasoning strategies [1], and consequently, they need different types of support.

The rest of the paper is divided as follows: In Section II, the syntax of the logic programs which we consider is introduced and the well-founded semantics is illustrated. In Section III, our main contribution is presented. Basically, we introduce all the components of the default argumentation context systems. In the last section, we outline our conclusions and future work.

II. BACKGROUND

In this section, some basic concepts of logic programs are presented. In particular, the syntax of extended normal logic programs is presented. For capturing the semantics of these programs, the well-founded semantics ([4]) is considered, by lack of space we omit its formal definition. We assume that the reader is familiar with basic background on Logic Programming with *negation as failure*. A good introduction to Logic Programming with negation as failure can be found in [5]

A. Normal Logic Programs

The language of propositional logic has an alphabet consisting of:

- (i) propositional symbols: p_0, p_1, \dots ;
- (ii) connectives : $\vee, \wedge, \leftarrow, \neg, \text{not}, \top$;
- (iii) auxiliary symbols : $(,)$;

in which \vee, \wedge, \leftarrow are 2-place connectives, \neg, not are 1-place connectives and \top is a 0-place connective. The propositional symbols, \top , and the propositional symbols of the form $\neg p_i$ ($i \geq 0$) stand for the indecomposable propositions, which we call *atoms*, or *atomic propositions*. Atoms negated by \neg will be called *extended atoms*. We will use the concept of atom without paying attention to whether it is an extended atom or not. The negation sign \neg is regarded as the so called *strong negation* by the Answer Set Programming's literature and the negation *not* as the *negation as failure*. A literal is an atom, a (called positive literal), or the negation of an atom *not a* (called negative literal).

A (propositional) extended normal clause, C , is denoted:

$$a \leftarrow b_1 \wedge \dots \wedge b_j \wedge \text{not } b_{j+1} \wedge \dots \wedge \text{not } b_{j+n} \quad (1)$$

where $j+n \geq 0$, a is an atom and each b_i ($1 \leq i \leq j+n$) is an atom. When $j+n = 0$ the clause is an abbreviation of $a \leftarrow \top$ such that \top is the propositional symbol that always evaluates to true. In a slight abuse of notation, we sometimes write the clause (1) as $a \leftarrow B^+ \wedge \text{not } B^-$, where $B^+ := \{b_1, \dots, b_j\}$ and $B^- := \{b_{j+1}, \dots, b_{j+n}\}$. An extended normal program

P is a finite set of extended normal clauses. When $n = 0$, the clause is called *extended definite clause*. An *extended definite logic program* is a finite set of extended definite clauses. By \mathcal{L}_P , we denote the set of atoms in the signature of P . Let $\text{Prog}_{\mathcal{L}}$ be the set of all normal programs with atoms from \mathcal{L} .

We will manage the strong negation (\neg) in our logic programs as it is done in Answer Set Programming (ASP) [5]. Basically, each atom of the form $\neg a$ is replaced by a new atom symbol a' which does not appear in the language of the program. In order not to allow inconsistent models from logic programs, a normal clause of the form $f \leftarrow a \wedge a' \wedge \text{not } f$ such that $f \notin \mathcal{L}_P$ is added.

Example 1: We illustrate a normal logic program with an example from the dementia domain (simplified due to space reasons). A summary of the clinical guidelines, which are used in the dementia example given here can be found in [6]. We use the following abbreviations:

<i>AD</i>	=	<i>Alzheimer's disease</i>
<i>DLB</i>	=	<i>Lewy body type of dementia</i>
<i>VaD</i>	=	<i>Vascular dementia</i>
<i>epiMem</i>	=	<i>Episodic memory dysfunction</i>
<i>fluctCog</i>	=	<i>Fluctuating cognition</i>
<i>fn</i>	=	<i>Focal neurological signs</i>
<i>prog</i>	=	<i>Progressive course</i>
<i>radVasc</i>	=	<i>Radiology exam shows vascular signs</i>
<i>slow</i>	=	<i>Slow, gradual onset</i>
<i>extraPyr</i>	=	<i>Extrapyramidal symptoms</i>
<i>visHall</i>	=	<i>Visual hallucinations</i>

By considering the previous abbreviations as propositional atoms, let P be a normal logic program formed by the following set of normal clauses.

- 1) $VaD \leftarrow fn \wedge \text{not } AD \wedge \text{not } DLB$
- 2) $VaD \leftarrow radVasc \wedge \text{not } AD \wedge \text{not } DLB$
- 3) $AD \leftarrow slow \wedge prog \wedge epiMem \wedge \text{not } VaD \wedge \text{not } DLB$
- 4) $DLB \leftarrow extraPyr \wedge visHall \wedge \text{not } fn$
- 5) $DLB \leftarrow fluctCog \wedge visHall \wedge \text{not } fn$
- 6) $DLB \leftarrow fluctCog \wedge extraPyr \wedge \text{not } fn$
- 7) $VaD \leftarrow fn \wedge radVasc$

These normal clauses will be considered for building arguments in the following sections.

B. Logic semantics

In this section, we present basic intuitions of 3-valued logic programming semantics. To this end, we present a basic definition of a 3-valued logic programming semantics.

Definition 1 (SEM [7]): For a normal logic program P , we define $\text{HEAD}(P) := \{a \mid a \leftarrow B^+ \wedge \text{not } B^- \in P\}$ — the set of all head-atoms of P . We define $\text{SEM}(P) = \langle P^{\text{true}}, P^{\text{false}} \rangle$, where $P^{\text{true}} := \{p \mid p \leftarrow \top \in P\}$ and $P^{\text{false}} := \{p \mid p \in \mathcal{L}_P \setminus \text{HEAD}(P)\}$. $\text{SEM}(P)$ is called a model of P .

Basically, the 3-valued model $SEM(P)$ of a given logic program P identifies three different classes of atoms:

- 1) Atoms which are considered *true* w.r.t. $SEM(P)$, i.e. atoms from \mathcal{L}_P which belong to P^{true} ;
- 2) Atoms which are considered *false* w.r.t. $SEM(P)$, i.e. atoms from \mathcal{L}_P which belong to P^{false} ;
- 3) Atoms which are considered *undefined* w.r.t. $SEM(P)$, i.e. atoms from \mathcal{L}_P which do not belong to $P^{true} \cup P^{false}$.

In the logic programming literature, there are different 3-valued logic programming semantics which have been intensively studied [8], [9]. Among these logic programming semantics, the Well-Founded Semantics (WFS) [4] satisfies well-expected properties about non-monotonic reasoning [9]. Indeed, WFS is a *well-behaved semantics* [9] and is regarded as an approximation of the Sable Model Semantics [10] which is the core the *Answer Set Programming* paradigm. An important computational property of WFS is that it is polynomial time computable. Nowadays, there are several solvers which can compute WFS in an efficient way: DLV⁴, SMOELS⁵, XSB⁶. In this paper, we will use the inference of WFS. By lack of space, we omit the formal definition of WFS; however, the reader can find the WFS's definition in [4] and use one of the mentioned solvers for computing WFS. Given a normal logic program, $WFS(P)$ will denote the well-founded model of P . Like $SEM(P)$, $WFS(P)$ is basically a 3-valued model. In order to illustrate $WFS(P)$, let us consider the following example:

Example 2: Let P be the following normal logic program:
 $b \leftarrow not\ a. \quad c \leftarrow not\ b. \quad c \leftarrow a.$

By using a WFS solver, we can see that $WFS(P) = \{\{b\}, \{a, c\}\}$. This means that the atom b is true according with $WFS(P)$; on the other hand, the atoms a and c are false according with $WFS(P)$.

In order to simplify to the presentation of some definitions (in the next sections), we introduce some notation. Let P be a normal logic program and $WFS(P) = \langle T, F \rangle$ be the well-founded model of P . Hence

- $P \models_{WFS^T} a$ if and only if $a \in T$;
- $P \models_{WFS^F} a$ if and only if $a \in F$

For instance, by considering the normal program P and $WFS(P)$ from Example 2, we can see that $P \models_{WFS^T} b$, $P \models_{WFS^F} a$ and $P \models_{WFS^F} c$.

III. DEFAULT ARGUMENTATION CONTEXT SYSTEMS

In this section, the idea of *Default Argumentation Context Systems* will be defined. To this end, two kinds argumentation frameworks will be defined: *deductive argumentation frameworks* and *abductive argumentation frameworks*. The idea is that given a set of observations a deductive argumentation framework, built from a deductive

knowledge base, will infer conclusions. On the other hand, given the conclusions of the deductive argumentation framework, an abductive argumentation framework, built from an abductive knowledge base, will support the conclusions of the given deductive argumentation framework. In order to merge the two kinds of argumentation framework, we will follow the ideal of *Argumentation Context Systems* which were introduced in [11]. We will start defining deductive argumentation frameworks.

A. Deductive Argumentation Frameworks

The structure of deductive argumentation frameworks (DAFs) will follow the well-known structure of *argumentation frameworks* which were introduced by Dung [12]. Hence, a DAF basically is a set of deductive arguments and a set of attacks between them. Therefore, let us start defining deductive arguments.

Definition 2 (Deductive argument): Let P be an extended normal logic program and $O \subseteq \mathcal{L}_P$ such that O is called observations. $A_D = \langle S, O', c \rangle$ is a deductive argument if the following conditions holds:

- 1) $S \cup O' \models_{WFS^T} c$,
- 2) $S \subseteq P$ such that S is a minimal set among the subsets of P satisfying 1,
- 3) $O' \subseteq O$ such that O' is a minimal set among the subsets of O satisfying 1.
- 4) $WFS(S \cup O') = \langle T, F \rangle$ such that $\nexists a \in \mathcal{L}_P$ and $\{a, \neg a\} \subseteq T$.

$A^D(P, O)$ denotes the set of deductive arguments built from P and O .

As we can observe in Definition 2, a deductive argument basically is a tuple of the form $\langle S, O', c \rangle$. The first condition of the definition suggests that $S \cup O'$ is the support of the argument and c is the claim of the argument. From conditions 2 and 3, we can guarantee that the support of the argument is the minimum information which can infer c by considering WFS. The last condition guarantees that this support is consistent.

Let us consider the following scenario in order to illustrate the definition of deductive arguments: An older person comes to the emergency room, after having fallen in her home. This has happened several times lately, and she cannot understand why and is getting very worried. In addition, she has difficulties in performing activities in her daily life, and it turns out that she has a state of dementia. However, this does not explain why she is falling, so the physician finds a reason to investigate further. The physical and cognitive examinations result in finding extrapyramidal symptoms (parkinsonism but without having a Parkinson's disease diagnosis), focal neurological symptoms, in addition to that her cognitive functions tend to fluctuate during the day. At this point the experienced physician is able to generate a set of hypothetical diagnoses containing the correct one, based on the information. The less experienced typically has difficulties using this approach to generate hypotheses [1], but may utilize

⁴<http://www.dbai.tuwien.ac.at/proj/dlv/>

⁵<http://www.tcs.hut.fi/Software/smodels/>

⁶<http://xsb.sourceforge.net/>

a decision support application, which behaves in the following way.

Example 3: Let P be the normal program introduced in Example 1 and $O = \{fn, extraPyr, fluctCog\}$. Some deductive arguments which one can build from P and O are:

$$\begin{aligned} Arg_D^1 &= \langle \{\}, \{fn\}, fn \rangle \\ Arg_D^2 &= \langle \{DLB \leftarrow fluctCog \wedge extraPyr \wedge not\ fn\}, \\ &\quad \{fluctCog, extraPyr\}, DLB \rangle \\ Arg_D^3 &= \langle \{VaD \leftarrow fn \wedge not\ AD \wedge not\ DLB\}, \\ &\quad \{fn\}, VaD \rangle \end{aligned}$$

From these arguments, we can believe that the given patient could be diagnosed with *Lewy body type dementia* (DLB) or *Vascular dementia* (VaD). However, these arguments are not final decisions. In order to have candidate decisions, we require to consider their disagreements and to select potential *acceptable* deductive arguments.

Once we have defined the structure of a deductive argument, the attack relation between these arguments is defined as follows:

Definition 3 (Attack relation between deductive arguments): Let $A = \langle S_A, O_A, c_A \rangle$, $B = \langle S_B, O_B, c_B \rangle$ be two deductive arguments, $WFS(S_A \cup O_A) = \langle T_A, F_A \rangle$ and $WFS(S_B \cup O_B) = \langle T_B, F_B \rangle$. We say that A attacks B if one of the following conditions holds:

- $a \in T_A$ and $\neg a \in T_B$.
- $a \in T_A$ and $a \in F_B$.

$At^D(S)$ denotes the set of attack relations between the deductive arguments which belong to a set of deductive arguments S .

Observe that the first condition of the definition is capturing the standard idea of *rebut* and the second condition is capturing the standard idea of *undercut*. Rebut and undercut are two well accepted ideas of disagreement between arguments in the argumentation theory [13].

Example 4: Let us consider the three deductive arguments which were introduced in Example 3. One can see the following relations of attack:

$$Arg_D^1 \text{ attacks } Arg_D^2 \quad Arg_D^2 \text{ attacks } Arg_D^3$$

Now we are in position for introducing the definition of a deductive argumentation framework.

Definition 4 (Deductive Argumentation Framework): Let P be an extended normal logic program and $O \subseteq \mathcal{L}_P$. A deductive argumentation framework is a tuple of the form $\langle \mathcal{A}^D(P, O), At^D(\mathcal{A}^D(P, O)) \rangle$.

As we can see, a deductive argumentation framework basically follows the structure of Dung's argumentation frameworks. The main difference is that a deductive argument has a structure which is based on logic programs with negation as failure and the inference of WFS. On the other hand, the attacks relation between deductive arguments is based on the inference of WFS.

It is quite easy to see that given the structure of a deductive argumentation framework, one can use an *extension-based argumentation semantics* [12] for selecting

sets of acceptable deductive arguments from a deductive argumentation framework.

Example 5: Let AR^D be the deductive arguments introduced in Example 3. Hence $DAF = \langle AR^D, At^D(AR^D) \rangle$ is a deductive argumentation framework. $At^D(AR^D)$ is composed by the two attack relations identified in Example 4. By considering the grounded semantics (introduced in [12]), we can see that $\{Arg_D^1, Arg_D^3\}$ is the grounded extension⁷ of DAF . By observing Arg_D^3 , a physician can believe that the given patient could has Vascular dementia (VaD). The question here is: *how can a physician validate this diagnosis?* We will give an answer to this question in the next sections.

An interesting property of WFS is that this logic programming semantics satisfies *relevance* [9]. This property takes importance from the argumentation point of view in order to show that the join of the pieces of knowledge which support each argument which belongs to an extension is *consistent*⁸. The only requirement of the given extension is that has to be conflict-free, *e.g.*, the extension does not contain two arguments which attack each other. Hence, we say that an extension-based argumentation semantics s satisfies conflict-freeness if each extension which is inferred by s is conflict-free.

Proposition 1: Let P be an extended normal logic program, $O \subseteq \mathcal{L}_P$, $DAF = \langle \mathcal{A}^D(P, O), At^D(\mathcal{A}^D(P, O)) \rangle$ be a deductive argumentation framework and s be an extension-based argumentation semantics which satisfies conflict-freeness. If $E \in s(DAF)$, $P_E = \{S \cup O \mid \langle S, O, c \rangle \in E\}$ and $C_E = \{c \mid \langle S, O, c \rangle \in E\}$ then the following conditions hold:

- 1) $P_E \models_{WFS^T} c$ such that $c \in C_E$
- 2) There is not $c \in \mathcal{L}_P$ such that $P_E \models_{WFS^T} c$ and $P_E \models_{WFS^T} \neg c$

B. Abductive Argumentation Frameworks

The other kind of argumentation frameworks that we will consider are the so called *abductive argumentation frameworks*. The novice physician in our example may have had difficulties finding hypothetical diagnoses using the deductive reasoning approach. Instead, he considers the dementia diagnoses which he is familiar with, and identifies their effects which match his observations.

Like deductive argumentation frameworks, abductive argumentation frameworks are based on logic programs and the inference of WFS. However, the logic programs which are considered for building abductive argumentation frameworks will be a particular class of logic programs which are called *Abductive Logic Programs*.

Definition 5 (Abductive Program): Let P an extended logic program. An abductive logic program is a pair $\langle P, H \rangle$ where the following conditions hold:

⁷An extension is a set of arguments which is suggested by an extension-based argumentation semantics.

⁸We say that a logic program P is consistent if there is not a model of P which contains a and $\neg a$.

- 1) $H \subset \mathcal{L}_P$, H will be called hypothesis.
- 2) P is an extended normal logic program such $HEAD(P) \cap H = \emptyset$.

This definition follows the ideas of abductive programs introduced in [14]. Hence, by considering this definition an abductive argument is defined as follows:

Definition 6 (Abductive Argument): Let $P_{Ab} = \langle P, H \rangle$ be an abductive logic program and O a set of atoms. An abductive argument with respect to an atom $a \in O$ is $A^{Ab}(a) = \langle S, E, a \rangle$ such that the following conditions holds:

- $S \cup E \models_{WFS^T} a$
- $S \subseteq P$, $E \subseteq H$ and both S, E are minimal amount the subsets of P and E respectively.

$\mathcal{A}^{Ab}(P_{Ab}, A)$ denotes the set of abductive arguments built from P_{Ab} and A .

As the deductive argument (see Definition 2), an abductive argument basically is a tuple of the form $\langle S, E, a \rangle$ in which we can find a support of the argument ($S \cup E$) and a claim a . The definition of an abductive argument also requests that the support of the argument has to be minimal. Unlike to deductive arguments which take observations as part of their support, abductive arguments take hypothesis as part of their support. Indeed, we can observe that an abductive argument explains an observation $a \in O$.

In the following example, we are going to illustrate abductive arguments.

Example 6: By considering the propositional atoms introduced in Example 1, let $P_{Ab} = \langle P', H \rangle$ be an abductive program such that $H = \{DLB, VaD, AD\}$, and P' the following set of normal clauses:

$extraPyr \leftarrow DLB.$
 $fluctCog \leftarrow DLB.$
 $visHall \leftarrow DLB.$
 $fn \leftarrow VaD.$
 $radVasc \leftarrow VaD.$
 $epiMem \leftarrow AD.$

These normal clauses were defined by re-interpreting the clinical guidelines in order to explore what information they give on causality, i.e., what can we expect to observe in an individual with a certain disease.

Let us consider the findings obtained in Example 3 $O = \{fn, extraPyr, fluctCog\}$ such that we want to find an explanation for each element of O by considering P_{Ab} . Hence, some abductive arguments from P_{Ab} and O are:

$Arg_1^{Ab} = \langle \{extraPyr \leftarrow DLB\}, \{DLB\}, extraPyr \rangle$
 $Arg_2^{Ab} = \langle \{fluctCog \leftarrow DLB\}, \{DLB\}, fluctCog \rangle$
 $Arg_3^{Ab} = \langle \{fn \leftarrow VaD\}, \{VaD\}, fn \rangle$

The first argument argues (Arg_1^{Ab}) that *extrapyramidal symptoms* (*extraPyr*) can be explained by *Lewy body dementia* (*DLB*). The last two argument have easy readings. Here the experienced physician is able to verify his hypotheses obtained using the deductive approach. The less experienced physician will limit his reasoning to the diagnoses he is familiar with, thus may miss the less common alternative

DLB and jump to the conclusion that VaD is the cause of the symptoms [1] which is an insufficiently informed decision.

Due to the aim of abductive arguments in our particular application, which is to support the claims of deductive arguments, we will not define a *particular* attack relation (disagreements) between abductive arguments. In this setting, abductive argumentation frameworks will be defined as follows:

Definition 7 (Abductive Argumentation Frameworks): Let $P_{Ab} = \langle P, H \rangle$ be an abductive logic program and O be a set of atoms. An abductive argumentation framework w.r.t. P_{Ab} and O is defined as follows: $\langle \mathcal{A}^{Ab}(P_{Ab}, O), At \rangle$ such that $At \subseteq \mathcal{A}^{Ab}(P_{Ab}, O) \times \mathcal{A}^{Ab}(P_{Ab}, O)$.

Like deductive argumentation frameworks, abductive argumentation frameworks follow the structure of Dung's argumentation frameworks. Hence one can use extension-based argumentation semantics for selecting acceptable abductive arguments from an abductive argumentation framework. Indeed, one can formulate a version of Proposition 1 for abductive argumentation frameworks.

Proposition 2: Let $P_{Ab} = \langle P, H \rangle$ be an abductive logic program, O be a set of atoms, $AAF(\mathcal{A}^{Ab}(P_{Ab}, O), At)$ be an abductive argumentation framework and s be an extension-based argumentation semantics which satisfies conflict-freeness. If $E \in s(AAF)$, $P_E = \{S \cup H \mid \langle S, H, c \rangle \in E\}$ and $C_E = \{c \mid \langle S, H, c \rangle \in E\}$ then the following conditions hold:

- 1) $P_E \models_{WFS^T} c$ such that $c \in C_E$
- 2) There is not $c \in \mathcal{L}_P$ such that $P_E \models_{WFS^T} c$ and $P_E \models_{WFS^T} \neg c$

C. Argumentation Context Systems

So far we have instantiated a deductive knowledge based into a deductive argumentation framework and an abductive knowledge base into an abductive argumentation framework. In order to support informed decision making (e.g., medical diagnosis in our running example with the two differently experienced physicians) these two argumentation frameworks need to be combined, or merged.

To this end, we will follow the approach of *Argumentation Context Systems* [11]; in particular, we will define the idea of *default argumentation context systems*. For this purpose, we introduce a definition of *context expressions* and *contexts* which are key features of argumentation context systems.

From here whenever we use only the word argument, it means that this argument can be either a deductive argument or an abductive argument.

Definition 8 ([11]): A context expression for a set of arguments AR and a set of values V has one of the following forms $(a, b \in AR; v, v' \in V)$:

$arg(a)/\overline{arg(a)}$	a is a valid/invalid argument
$att(a,b)/\overline{att(a,b)}$	(a,b) is valid/invalid attack
$a > b$	a is strictly preferred to b
$val(a,v)$	the value of a is v
$v > v'$	value v is strictly better than v'
$mode(r)$	the reasoning mode $r \in \{skip, cred\}$
$sem(s)$	s is a chosen argumentation semantic [12]

A context C is a set of context expressions.

One can see that a context defines different aspects w.r.t. a set of arguments. For instance:

- a) preferences between arguments;
- b) validity/invalidity of specific arguments (and attacks);
- c) a reasoning mode (either skeptical or credulous); and
- d) an extension-based argumentation semantics (grounded, stable, preferred, *etc.*) [12].

Hence, by considering a given context one can modify a (deductive/abductive) argumentation framework as follows:

Definition 9: Let $AF = \langle AR, att \rangle$ be a (deductive/abductive) argumentation framework, V be a set of values and C be a context for AR and V . The C^\top -modification of AR is the (deductive/abductive) argumentation framework $AF^C = \langle AR^C, att^C \rangle$, where

- $AR^C = AR \cup \{\{\}, \{\}, \top\}$
- att^C is the smallest relation satisfying the following conditions:
 - 1) if $att(a,b) \in C$, then $(a,b) \in att^C$,
 - 2) if $(a,b) \in att$, $att(a,b) \notin C$, and $b \not\prec_C a$, then $(a,b) \in att^C$,
 - 3) if $att(a,b) \in C$ or $(arg(b) \in C \wedge (a,b)att^C)$ then $(\{\}, \{\}, \top, a) \in att^C$.

Let observe that the new argument $\{\{\}, \{\}, \top\}$, which is non-attackable, basically is defeating invalid arguments as well as attackers of valid arguments.

Following with *argumentation context systems*, we will consider the C^\top -modification for defining the sets of acceptable arguments.

Definition 10: Let $AF = \langle AR, att \rangle$ be a (deductive/abductive) argumentation framework, V be a set of values and C be a context for AR and V such that $sem(s) \in C$, $mode(m) \in C$. A set of arguments $S \subseteq AR$ is an acceptable C^\top -extension for AF , if either:

- $m = cred$ and $S \cup \{\{\}, \{\}, \top\}$ is a s-extension of AF^C , or
- $m = skip$ and $S \cup \{\{\}, \{\}, \top\}$ is the intersection of all s-extensions of AF^C .

where s-extension is an extension (a set of arguments) according to the extension-based argumentation semantics s .

In order to define *default argumentation context systems*, let us introduce the so called *bridge rules*. To this end, let us define the following notation: Let AR be a set of arguments and Σ be a set of atomic symbols of the same cardinality

of AR . \bullet is a bijective function from AR onto Σ . We shall denote the image of $a \in AR$ under \bullet as a^\bullet . A straightforward generalization of \bullet over AR is: $AR^\bullet = \{a^\bullet | a \in AR\}$.

Given a set of argument AR , a bridge rule is a normal clause of the form:

$$s \leftarrow a_1^\bullet \wedge \dots \wedge a_n^\bullet \wedge \text{not } a_{n+1}^\bullet \wedge \dots \wedge \text{not } a_m^\bullet \quad (2)$$

where s is a context expression and $a_i \in AR (1 \leq i \leq m)$.

By considering bridge rules, we define a mediator as follows:

Definition 11: Let AF_1 and AF_2 be arbitrary (deductive/abductive) argumentation frameworks. A mediator for AF_1 based on AF_2 is a tuple of the form: $Med = \langle E, R_1 \rangle$, where E is a set of context expressions for AF_1 and R_1 is a set of bridge rule of the form (2) such that s is a context expression for AF_1 and the arguments are from AF_2 .

Now that we have defined mediators between argumentation frameworks, a module M is a tuple of the form $\langle AF, M \rangle$ such that AF is a deductive/abductive argumentation framework and M is a mediator for AF based on a given deductive/abductive argumentation framework.

A default argumentation context system is a pair of modules which is defined as follows:

Definition 12: A default argumentation context system (DACS) is a tuple of the form: $DAF = \langle M_1, M_2 \rangle$, where $M_1 = \langle AF_1, Med_1 \rangle$, $M_2 = \langle AF_2, Med_2 \rangle$, AF_1 is a deductive argumentation framework, AF_2 is an abductive argumentation framework, Med_1 is based on AF_2 and Med_2 is based on AF_1 .

Given the implicit dependency between M_1 and M_2 , one can see that given sets of acceptable arguments for M_1 , the mediators define the consistent *acceptable context* for M_2 and viceversa.

In order to define the semantics of default argumentation context systems, the acceptable states of a given default argumentation context system will be defined as follows.

Definition 13: Let $DACS = \langle M_1, M_2 \rangle$ be a default argumentation context System.

- A state of $DACS$ is a function S that assign each $M_i = \langle AF_i, Med_i \rangle$ a pair $S(M_i) = \langle Acc_i, C_i \rangle$ of a subset Acc_i of arguments of AF_i and a set C_i of context expressions for AF_i , $i \in \{1, 2\}$.
- A state S is stable if (i) each Acc_i is an acceptable C^\top -extension of AF_i and C_i is an acceptable context for M_i , $i \in \{1, 2\}$.

In order to define a *default argumentation context framework for our running example*, we are going to introduce some notations: Let P be a normal logic program, $P_{Ab} = \langle P', H \rangle$ be an abductive program and $O \subseteq \mathcal{L}_P$. If $a^d \in \mathcal{A}^D(P, O)$, then:

$$h(a^d) := a^{Ab} \text{ if } \begin{aligned} &a^{Ab} \in \mathcal{A}^{Ab}(P', O), \\ &a^{Ab} = \langle S, H, c' \rangle, c \in H \text{ and } c' \in O' \end{aligned}$$

Example 7: Let $DAF = \langle AR^D, At^D(AR^D) \rangle$ be the deductive argumentation framework introduced in Example 5 and AR^{Ab} be the set of abductive arguments introduced in

Example 6. One can see that $AAF = \langle AR^{Ab}, \{\} \rangle$ is an abductive argumentation framework.

Let R be the set of bridge rules $\{arg(a) \leftarrow b \mid a \in AR^D, b \in AR^{Ab} \text{ and } h(a) = b\}$. We can see that $R = \{arg(Arg_D^2) \leftarrow Arg_1^{Ab}, arg(Arg_D^2) \leftarrow Arg_2^{Ab}, arg(Arg_D^3) \leftarrow Arg_3^{Ab}\}$. An intuitive reading of the first bridge rule suggests that the abductive argument Arg_1^{Ab} supports the conclusion of the deductive argument Arg_D^2 . The same reading can be done with the rest of bridge rules.

Let $Med_1 = \langle \{sem(grounded), mode(skep)\}, R \rangle$ and $Med_2 = \langle \{sem(grounded), mode(skep)\}, \{\} \rangle$ be two mediators. Hence, a default argumentation context system $DACS_{running}$ of our running example can be: $DACS_{running} = \langle \langle DAF, Med_1 \rangle, \langle AAF, Med_2 \rangle \rangle$. The unique stable state of $DACS_{running}$ suggests that $Acc_1 = \{Arg_D^3\}$ and $Acc_2 = AR^{Ab}$. An interpretation of this stable state suggests that according with the default argumentation context system, the diagnosis of *Vascular dementia* (Vad) is supported by the deductive argumentation framework and the abductive argumentation framework. Let us remember that *Vad* was the conclusion in Example 5. However, this conclusion is now validated by Arg_3^{Ab} from AAF .

It is known that the computational cost of extension-based argumentation semantics is high. Indeed the computational complexity of the decision problems of the extension-based argumentation semantics range from NP-complete to $\Pi_2^{(p)}$ -complete [15]. One of the efficient extension-based argumentation semantics which exists is the grounded semantics. Hence, the use of the grounded semantics in each module of a default argumentation context system (DACS) implies to compute the stable state of DACS in a efficient way.

Proposition 3: Let $DACS = \langle M_1, M_2 \rangle$ be a default argumentation context system. If M_1 and M_2 use the grounded semantics, then the acceptable states of $DACS$ are computable in polynomial time.

IV. CONCLUSIONS AND FUTURE WORK

We have explored argumentation context systems in order to support informed decision making with heterogeneous knowledge bases. To this end, we have introduced two kinds of argumentation frameworks: *deductive* argumentation frameworks and *abductive* argumentation frameworks. These argumentation frameworks are based on logic programs with negation as failure and WFS. For merging these argumentation frameworks, Default Argumentation Context Systems have been introduced and exemplified by a medical diagnostic example.

An interesting property of WFS is that this logic programming semantics satisfies *relevance* [9]. This property takes importance from the argumentation point of view in order to show that the join of the pieces of knowledge which support each argument which belongs to an extension is *consistent*. Hence, we showed that the knowledge which belongs to a set of either deductive or abductive arguments is

consistent if this set of arguments belongs to an extension of an argumentation semantics which satisfies *conflict-free* (Proposition 1 and Proposition 2).

We also showed that by considering particular argumentation semantics as *the grounded semantics*, one can infer *collective acceptable states* from a default argumentation context system in polynomial time (Proposition 3).

We are particularly interested in supporting medical diagnosis in the demential domain. Hence, we argue that our approach *mimics* the reasoning process of an expert physician. In this context, we can say that the novice physician is able to use only fragments of the abductive knowledge base, which limits the quality of his assessments. By contrast, the expert physician is able to use both the deductive and abductive knowledge bases, which contributes to a holistic perspective on a patient case. Since the Default Argumentation Context Systems mimic the expert's reasoning, they may guide the novice physician into a reasoning process which generates more informed decisions at the same time as the physician develops his skills in diagnostic reasoning.

The quality of the informed decisions supported by the Default Argumentation Context Systems could be improved even further by integrating possibilistic values attached to hypotheses and preferences among knowledge sources. This will be our main focus for future work. Indeed, we have argued that the declarative specifications of medical guidelines require rich specification languages which could capture uncertain and incomplete information [16], [17].

ACKNOWLEDGMENT

This research is partly funded by VINNOVA (Sweden's innovation agency) and the Swedish Brain Power.

REFERENCES

- [1] V. L. Patel, D. R. Kaufman, and J. F. Arocha, "Emerging paradigms of cognition in medical decision-making," *Journal of Biomedical Informatics*, vol. 35, no. 1, pp. 52–75, 2002.
- [2] H. Lindgren and P. Eklund, "Differential diagnosis of dementia in an argumentation framework," *Journal of Intelligent and Fuzzy Systems*, vol. 17, no. 4, pp. 387–394, 2006.
- [3] G. Brewka, T. Eiter, and M. Fink, "Nonmonotonic multi-context systems: A flexible approach for integrating heterogeneous knowledge sources," in *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, ser. Lecture Notes in Computer Science, vol. 6565. Springer, 2011, pp. 233–258.
- [4] A. V. Gelder, K. A. Ross, and J. S. Schlipf, "The well-founded semantics for general logic programs," *Journal of the ACM*, vol. 38, no. 3, pp. 620–650, 1991.
- [5] C. Baral, *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge: Cambridge University Press, 2003.
- [6] J. O'Brien, D. Ames, and A. Burns, Eds., *Dementia*. Arnold, 2000.
- [7] J. Dix, M. Osorio, and C. Zepeda, "A general theory of confluent rewriting systems for logic programming and its applications," *Ann. Pure Appl. Logic*, vol. 108, no. 1-3, pp. 153–188, 2001.
- [8] J. Dix, "A classification theory of semantics of normal logic programs: I. strong properties," *Fundam. Inform.*, vol. 22, no. 3, pp. 227–255, 1995.
- [9] —, "A classification theory of semantics of normal logic programs: II. weak properties," *Fundam. Inform.*, vol. 22, no. 3, pp. 257–288, 1995.
- [10] M. Gelfond and V. Lifschitz, "The Stable Model Semantics for Logic Programming," in *5th Conference on Logic Programming*, R. Kowalski and K. Bowen, Eds. MIT Press, 1988, pp. 1070–1080.

- [11] G. Brewka and T. Eiter, "Argumentation Context Systems: A Framework for Abstract Group Argumentation," in *Logic Programming and Nonmonotonic Reasoning (LPNMR 2009)*, ser. Lecture Notes in Computer Science, vol. 5753. Springer, 2009, pp. 44–57.
- [12] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–358, 1995.
- [13] H. Prakken and G. A. W. Vreeswijk, "Logics for defeasible argumentation," in *Handbook of Philosophical Logic*, 2nd ed., D. Gabbay and F. Günthner, Eds. Dordrecht/Boston/London: Kluwer Academic Publishers, 2002, vol. 4, pp. 219–318.
- [14] C. Sakama and K. Inoue, "An abductive framework for computing knowledge base updates," *TPLP*, vol. 3, no. 6, pp. 671–713, 2003.
- [15] P. E. Dunne, "Computational properties of argument systems satisfying graph-theoretic constraints," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 701–729, 2007.
- [16] J. C. Nieves, M. Osorio, and U. Cortés, "Semantics for possibilistic disjunctive programs," *Theory and Practice of Logic Programming*, vol. 13, pp. 33–70, 0 2013. [Online]. Available: http://journals.cambridge.org/article_S1471068411000408
- [17] J. C. Nieves and H. Lindgren, "Possibilistic Nested Logic Programs," in *Technical Communications of the 28th International Conference on Logic Programming (ICLP'12)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), A. Dovier and V. S. Costa, Eds., vol. 17. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 267–276. [Online]. Available: <http://drops.dagstuhl.de/opus/volltexte/2012/3628>