Using First-Order Logic to Represent Clinical Practice Guidelines and to Mitigate Adverse Interactions

Martin Michalowski
Adventium Labs, USA
martin.michalowski@adventiumlabs.com

Szymon Wilk
Poznan University of Technology, Poland
szymon.wilk@cs.put.poznan.pl

Xing Tan, Wojtek Michalowski
University of Ottawa, Canada
{tan,wojtek}@telfer.uottawa.ca

Daniela Rosu
University of Toronto, Canada
drosu@cs.toronto.edu

Abstract

Clinical practice guidelines (CPGs) were originally designed to help with evidence-based management of a single disease and such a single disease focus has impacted research on CPG computerization. This computerization is mostly concerned with supporting different representation formats and identifying potential inconsistencies in the definitions of CPGs. However, one of the biggest challenges facing physicians is the personalization of multiple CPGs to comorbid patients. Various research initiatives propose ways of mitigating adverse interactions in concurrently applied CPGs, however, there are no attempts to develop a generalized framework for mitigation that captures generic characteristics of the problem while handling nuances such as precedence relationships. In this paper we present our research towards developing a mitigation framework that relies on a first-order logic-based representation and related theorem proving and model finding techniques. The application of the proposed framework is illustrated with a simple clinical example.

1 Introduction

A clinical practice guideline (CPG) codifies the evidence-based best practice in prescribing the most appropriate disease-specific therapy to patients, subject to available patient data and possible diagnoses (Rosenfeld and Shiffman 2009). Since the scope of each guideline is limited to a single disease, the evidence-based management of a comorbid patient according to the recommendations concurrently coming from multiple CPGs is difficult and can result in inconsistent and potentially harmful therapies. Often times the derivation of a combined therapy directly from the guidelines (even for properly diagnosed comorbid conditions) is incorrect due to adverse interactions between the treatments associated with individual therapies. These interactions manifest directly as contradictory recommendations (e.g., use of steroids is recommended by one CPG and prohibited by the other), or they may correspond to drug-drug or drug-disease adverse interactions resulting in actions that cannot be taken concurrently.

As a matter of fact, concurrent application of two or more CPGs is challenging - it requires designing a sophisticated mechanism for identifying and eliminating potential redundancy in the tests or procedures, identifying contradictions (direct adverse interactions), and for managing discordance (indirect, drug-drug or drug-disease interactions) (Sittig et al. 2008). As such, it is believed that executing multiple CPGs concurrently requires a new, "combinatorial, logical, or semantic" methodological approach (Fox et al. 2010).

Our previous research (Wilk et al. 2013; Michalowski et al. 2013a; 2013b) proposes such an approach by introducing and formally defining logical models of CPGs and developing a mitigation algorithm that operates on these models. The algorithm relies on secondary clinical knowledge (i.e., knowledge that goes beyond the primary knowledge encoded in CPGs and that comes from domain experts, textbooks, or repositories of clinical evidence) that is encoded as interaction and revision operators. The operators characterize adverse interactions associated with the concurrent application of CPGs and describe revisions to logical models required to address these interactions. The algorithm employs the constraint logic programming (CLP) paradigm to efficiently solve the logical models where a solution represents a combined and personalized therapy free of adverse interactions.

In the research described here, we move further towards developing a general framework for mitigation by enriching the representation of CPGs using first-order logic (FOL) theories and relying on theorem proving and model finding techniques to process these theories. This expansion is dictated by the following limitations of our previous research:

- Restricted expressive power of the CLP-based approach that does not allow for explicit representation of properties of objects (e.g., a dosage associated with a specific CPG action) and relationships between objects (e.g., precedence between CPG actions),
- Limited interpretability of solutions returned by CLP solvers and consequently the need to assign real-world semantics to truth-value assignment of the propositional symbols in the CLP-based model.

FOL significantly improves the expressiveness of our approach by introducing predicates to represent properties and relationships in the domain (in fact, relationships are only first-order definable). Moreover, predicates impose semantics on solutions, facilitating their interpretation from a clin-

ical perspective.

This paper is organized as follows. First we present the foundations of FOL and theorem proving and model finding that are relevant to our research. Next we describe the proposed framework - we start with the underlying FOL theories and then present an overview of the mitigation process. We proceed with a simple clinical example that illustrates the application of the framework. Finally, we finish with a brief review of related work and provide conclusions and directions for our future research.

2 Background

2.1 Foundations of FOL

The formal language of FOL relies on *logical* and *non-logical symbols*. The logical symbols (connectives, quantifiers, variables) are those that have a fixed meaning in a language. The non-logical symbols are those that have an application-dependent meaning (e.g., symbols needed to represent a CPG in FOL) and they are further categorized into *function* and *predicate* symbols. Each non-logical symbol has an *arity*, indicating how many arguments it requires. A function symbol with arity 0 is called a *constant* and a predicate symbol with arity 0 is called a *propositional symbol*.

FOL allows for two types of syntactic expressions: *terms* (made of variables, constants and functions) and *formulas* (composed of terms, predicates and connectives). Formulas with variables bounded by quantifiers and formulas without variables (i.e., grounded formulas) are called *sentences*. A FOL *theory* D is a collection of sentences. An *interpretation* I (sometimes called a structure) in FOL is defined as a triple:

$$I =< I_{domain}, I_{predicate}, I_{function} >,$$

where

- I_{domain} is any nonempty set of objects under consideration called the domain of the interpretation,
- $I_{predicate}$ is a set of interpretation mappings over I_{domain},
- $I_{function}$ is a set of functions over I_{domain}.

Mappings from $I_{predicate}$ assign meaning to the predicate symbols as follows: for every predicate symbol P of arity n, $I[P] \in I_{predicate}$ is an n-ary relation over I_{domain}, that is $I[P] \subseteq I_{domain} \times \ldots \times I_{domain}$. Mappings from $I_{function}$ assign meaning to the function symbols as follows: for every function symbol F of arity n, $I[F] \in I_{function}$ is an n-ary function over I_{domain}, that is $I[F] \in [I_{domain} \times \ldots \times I_{domain} \rightarrow I_{domain}]$. Given an interpretation I, we can check which sentences of a FOL theory D are true and which are false according to this interpretation. If sentence $\phi \in D$ is true given I, then we write it formally as $I \models_m \phi$. Moreover, if I satisfies all sentences in D, then it is called a *model* for theory D and formally it is denoted as $I \models_m D$.

2.2 Theorem Proving and Model Finding

There are three fundamental questions that are associated with FOL theories:

1. Is a given theory consistent?
2. What is a model for a consistent theory?
3. What are logical consequences (implications) of a consistent theory?

A FOL theory D is *consistent* (or satisfiable) iff there exists at least one model of this theory. The question of the consistency of D can be answered using *theorem proving* (Pavlov, Schukin, and Cherkasova 2013) that employs automatic reasoning (the resolution method) to construct a proof for D. However, theorem proving techniques provide only a binary answer to the consistency question and no model is directly returned even if it exists (i.e., when the answer is positive). In order to answer the question about a model for a consistent theory, one needs to use *model finding* techniques that can be considered as a special case of the constraint satisfaction problem (Zhang and Zhang 2013) where possible interpretations are generated until a model is found.

The logical consequences question is translated into checking if a FOL theory D entails sentence ϕ (or ϕ is a logical consequence of D). Formally, we say D *entails* ϕ, written as $D \models \phi$, iff, for every interpretation I such that $I \models_m D$, we have $I \models_m \phi$. In other words, we say D *entails* ϕ (or ϕ can be deduced from D) if ϕ is satisfied by all models for D. The entailment $D \models \phi$ can be translated into checking whether a new theory $D \cup \{\neg\phi\}$ is not consistent. This means that theorem proving techniques can equivalently be used to check for logical entailments of a theory D.

3 Methodology

Using FOL in a framework for the mitigation of concurrently applied CPGs relies on four key components that are listed below and described in the following sections:

1. A vocabulary used to construct the FOL theory describing a particular mitigation problem (further referred to as to *combined mitigation theory*),
2. A combined mitigation theory composed of individual theories that describe various aspects of the mitigation problem,
3. A set of operators that encode the secondary knowledge needed to identify and address adverse interactions associated with the combined mitigation theory,
4. A mitigation algorithm that controls the application of operators to the combined mitigation theory.

3.1 Vocabulary

Following our previous work, we assume a CPG is represented as an *actionable graph* (AG) (Wilk et al. 2013). An AG is a directed graph composed of three types of nodes - *context*, *action*, and *decision*, and arcs that represent transitions between nodes. A context node defines an entry point and indicates the disease associated with the CPG, an action node indicates a clinical action that needs to be executed, and a decision node indicates a selection from several alternative choices and allows for conditional branching. The vocabulary of our FOL-based approach is composed of constants (denoted with upper case letters), variables (denoted

Predicate	Description
node(x)	*x* is a node in AG
action(x)	*x* is an action node in AG
decision(x)	*x* is a decision node in AG
executed(x)	action node *x* is executed
value(x, v)	value *v* is associated with decision node *x*
dosage(x, n)	action node *x* is characterized by medication dosage *n*
directPrec(x, y)	node *x* directly precedes node *y* (there is an edge from *x* to *y*)
prec(x, y)	node *x* precedes node *y* (there is a path from *x* to *y*)
disease(d)	*d* is a disease to be managed
diagnosed(d)	the patient has been diagnosed with disease *d*

Table 1: Defined predicates

with lower case letters) and predicates. The predicates used in the mitigation problem are listed in Table 1. We note there is no predicate corresponding to a context node, as information embedded in this node is provided by the predicate *diagnosed(d)*.

3.2 Combined Mitigation Theory

We use the vocabulary to construct a combined mitigation theory. Formally, this combined theory D_{comb} is defined as a triple:

$$D_{comb} =< D_{common}, D_{cpg}, D_{pi} >,$$

where D_{common} is a theory that axiomatizes the universal characteristics of CPGs as part of a FOL representation. It is the common (shared and reusable) component of all mitigation theories and it contains the following axioms:

- $\forall x, y\ directPrec(x, y) \Rightarrow prec(x, y)$ - association between precedence and direct precedence,
- $\forall x, y,\ directPrec(x, y) \Rightarrow \neg directPrec(y, x)$ - asymmetry of direct precedence,
- $\forall x, y, z,\ directPrec(x, y) \wedge directPrec(y, z) \Rightarrow directPrec(x, z)$ - transitivity of direct precedence,
- $\forall x, y, z\ prec(x, y) \wedge prec(y, z) \Rightarrow prec(x, z)$ - transitivity of precedence,
- $\forall x, y,\ prec(x, y) \Rightarrow \neg prec(y, x)$ - asymmetry of precedence to ensure a strict partial order,
- $\forall x\ node(x) \Rightarrow (action(x) \wedge \neg decision(x)) \vee (\neg action(x) \wedge decision(x))$ - ensures that a node cannot be simultaneously an action and decision node,
- $\forall x, action(x) \Rightarrow node(x), \forall x, decision(x) \Rightarrow node(x)$ - all nodes are either actions or decisions,
- $\forall x, n\ dosage(x, n) \Rightarrow action(x)$ - ensures that only an action node can be characterized with medication dosage,
- $\forall x, v\ value(x, v) \Rightarrow decision(x)$ - ensures that only a decision node can be characterized by a value,
- $\forall d\ diagnosed(d) \Rightarrow disease(d)$ - ensures that the diagnosed disease is the same as the disease to be managed.

D_{cpg} is a union of theories, each theory representing a single AG (and thus the underlying CPG) that are being applied to a comorbid patient:

$$D_{cpg} = D_{cpg}^{d_1} \cup D_{cpg}^{d_2} \cup \ldots \cup D_{cpg}^{d_k},$$

where $D_{cpg}^{d_i}$ is the theory that describes the AG associated with disease d_i by enlisting all nodes and paths, giving information about precedence between nodes and providing information on dosages associated with selected action nodes. Because of axioms in D_{common} it is sufficient to define only direct precedence between nodes (*directPrec* predicate) - precedence between nodes represented with the *prec* predicate is derived automatically.

D_{pi} is the theory that describes available patient information. It contains sentences representing patient data, including results of tests and examinations, or indicating already prescribed therapies and procedures.

3.3 Interaction and Revision Operators

Interaction and revision operators were introduced in our previous research (Wilk et al. 2013). Here we reformulate them to account for the FOL-based representation and to enhance their capabilities. For example using a FOL-based approach enables the mitigation framework to represent a revision operator that specifies multiple operations.

An interaction operator IO^k encodes knowledge about indirect adverse interactions (usually drug-drug or drug-disease) and formally it is defined as

$$IO^k =< \alpha^k >,$$

where α^k is a sentence (constructed with the vocabulary described in Section 3.1) describing a specific indirect interaction. Checking whether IO^k is applicable to D_{comb} (or in other words, if the interaction represented by IO^k occurs in D_{comb}) is the entailment problem $D_{comb} \models \alpha^k$.

A revision operator encodes knowledge about the revisions that need to be introduced to the theory D_{cpg} in order to address encountered interactions (both direct and indirect). In layman terms, it describes changes that need to be introduced to concurrently applied CPGs. Formally a revision operator RO^k is defined as

$$RO^k =< \beta^k, Op^k >,$$

where β^k is a logical sentence that defines the applicability of the operator to the theory D_{cpg}, and Op^k describes the revisions introduced by RO^k. In particular, Op^k is a set of n pairs of formulas $< \varphi_i^k, \phi_i^k > (i = 1 \ldots n)$ that define a single operation within the operator. As already stated, these

operations are applied only to D_{cpg}, so other components of D_{comb} are protected from unwanted revisions. For example, D_{pi} is never modified thus patient information is never inadvertently changed. The pairs of formulas are interpreted as follows (where $\emptyset$ indicates an empty formula):

- $< \varphi_i^k, \emptyset >$ means that φ_i^k is removed from any sentence in D_{cpg} where it appears,
- $< \emptyset, \phi_i^k >$ means that ϕ_i^k is added as a new sentence to D_{cpg},
- $< \varphi_i^k, \phi_i^k >$ means that φ_i^k is replaced by ϕ_i^k in any sentence in D_{cpg} where it appears.

It is possible to use unbounded variables in φ_i^k and ϕ_i^k and these are interpreted as "wildcards" that are bound to a constant specific to a patient encounter. For example, one can define an operation that increases the dosage of a medication by a given amount. Moreover, checking the applicability of RO^k to D_{comb} is analogous to checking the applicability of IO^k and translates into the entailment problem $D_{comb} \models \beta^k$.

3.4 Mitigation Algorithm

The algorithm consists of two phases and it is outlined in Figure 1. The first phase involves mitigating direct adverse interactions. Their identification translates into checking the consistency of the D_{comb} theory (note that in order to check for consistency and entailment we need to create a temporary theory that is a union of all three components in D_{comb}). If the theory is consistent, then it indicates there are no direct interactions and the algorithm passes to the second phase. Otherwise, the theory D_{comb} (specifically its D_{cpg} component) needs to be revised using applicable revision operators. Since D_{comb} is inconsistent, entailment cannot be used to find applicable revision operators as entailment problems can only be formulated over a consistent theory. Instead we identify actions shared across theories in D_{cpg} that result in direct interactions and use them to check the applicability of RO^k. The algorithm may stop here, reporting a failure to indicate that D_{comb} is still inconsistent, if it has failed to address the encountered direct interaction.

The second phase identifies and addresses indirect adverse interactions. It starts by identifying applicable interaction operators (for operator IO^k this translates to checking the entailment $D_{comb} \models \alpha^k$). If there is no applicable operator, then this means that there are no indirect interactions or they have been already addressed and the algorithm finds a model for D_{comb}. This model is equivalent to a solution in the CLP-based mitigation framework, and using its $I_{predicate}$ component it is possible to construct a personalized combined therapy for a patient. This combined therapy highlights the clinical actions to be taken (*executed* and *dosage* predicates) along with the order in which they should be carried out (*prec* and *directPrec* predicates) and includes the assumptions made about the patient's state (*value* predicates). Note that the combined therapy contains only these predicates that have not been provided as part of D_{pi}, thus it is focused on future (suggested) actions and possible (assumed) patient state.

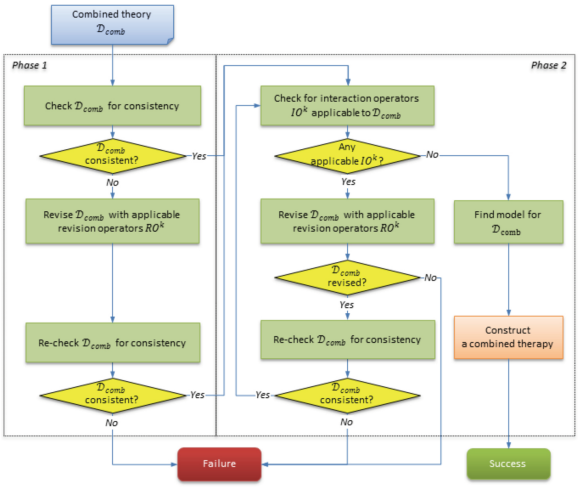

Figure 1: Outline of the mitigation algorithm.

On the other hand, if direct interactions exist (there is at least one IO^k applicable to D_{comb}), the algorithm attempts to revise D_{comb} using applicable revision operators, where checking applicability of an operator RO^k is an entailment problem ($D_{comb} \models \beta^k$). There is an additional explicit check if D_{comb} has been revised to avoid indefinite loops if there is no applicable RO^k. If the revised D_{comb} is consistent, then the algorithm checks again for an applicable IO^k, otherwise it fails. This loop is repeated until there are no more applicable interaction operators.

In our previous research we assumed that an interaction had to be addressed by a single applicable revision operator. In this framework we relax this assumption and allow for more complex adverse interactions that may need to be mitigated by multiple revision operators. Further, the implementation of the mitigation algorithm involves a number of software tools that were developed for FOL theories. In this research we are using Prover9 (McCune 2005) to check consistency of all theories and to execute the entailment required for the identification and use of the operators. Moreover, we are using a model finding technique implemented in Mace4 (McCune 2005) that returns a model on top of a theory that has been verified as a consistent one.

4 Illustrative Example

In this section we illustrate our proposed FOL-based mitigation framework using the simple clinical case also used in (Wilk et al. 2013). The purpose of using the same example is to show how the methodology proposed here extends our earlier research. According to this example, a patient that is treated for a duodenal ulcer (DU) experiences an episode of transient ischemic attack (TIA). AGs used in this example are derived from the guidelines published by the National Institute for Health and Clinical Excellence, UK (NICE) (NIC 2012) and they have been simplified to include only the relevant action and decision nodes.

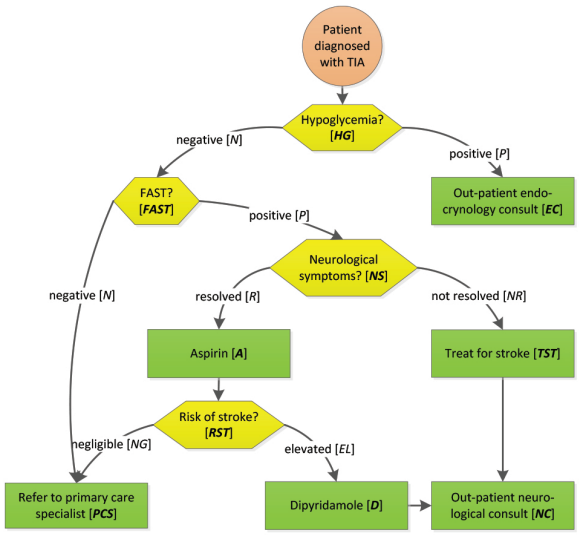

Figure 2: Actionable graph for TIA (AG_{TIA}).

4.1 Actionable Graphs

For illustrative purposes we show in Figure 2 the AG for TIA only. In this figure the context node is indicated with a circle, decision nodes are indicated with diamonds, and action nodes with rectangles. The figure also labels constants associated with specific nodes and corresponding to alternative choices - they are given in square brackets after node and choice descriptions.

4.2 Theories

The AGs are converted into the respective theories, D_{cpg}^{DU} for DU and D_{cpg}^{TIA} for TIA. For illustrative purposes, D_{cpg}^{TIA} is shown in Figure 3. Note that *DU*, *HP*, *P*, *TIA*, *HG*, *N*, etc. are all FOL constants. As can be seen, this representation captures precedence relationships and attaches semantics to each node. All paths in the corresponding AG are described using a single sentence (a disjunction of conjunctions, where each conjunction corresponds to a single path). Each path contains formulas with the negated *executed* predicate to indicate these actions are not executed for a given path.

4.3 Operators

One interaction and two revision operators associated with the clinical scenarios are discussed below.

$$IO^1 =< \alpha^1 >,$$

$$\alpha^1 = diagnosed(DU) \wedge executed(A) \wedge \neg executed(PPI).$$

IO^1 represents a drug-disease interaction (the increased risk of bleeding) that occurs when a DU patient is given aspirin (A) without a proton-pump inhibitor (PPI).

$$RO^1 =< \beta^1, Op^1 >,$$

$$\beta^1 = diagnosed(DU) \wedge executed(A) \wedge \neg executed(D),$$

$$Op^1 = \{< executed(A),\ executed(CL) >\}$$

RO^1 is applicable to a patient diagnosed with DU who has been prescribed aspirin (A) but has not been prescribed dipyridamole (D). In such cases, the patient is taken off of aspirin and prescribed clopidogrel (CL).

$$RO^2 =< \beta^2, Op^2 >,$$

$$\beta^2 = diagnosed(DU) \wedge executed(A) \wedge executed(D),$$

$$Op^2 = \{< \emptyset,\ executed(PPI) >, < dosage(A, x),\ dosage(A, x - 50) >\}$$

RO^2 is applicable to a patient diagnosed with DU who has been prescribed aspirin (A) and dipyridamole (D). In such cases, the patient is also prescribed a proton-pump inhibitor (PPI) and the dosage of aspirin (A) is reduced by 50 milligrams (mg).

4.4 Scenario 1: No Adverse Interactions

In this scenario we assume a patient suffering from DU who has tested positive for H.pylori (HP) and is undergoing eradication therapy (ET), on presentation to the emergency department with TIA symptoms, has tested negative for hypoglycemia (HG) and the result of FAST test (FAST) is negative. The theory D_{pi} describing this patient is:

$$diagnosed(DU).value(HP, P).executed(ET).$$
$$diagnosed(TIA).value(HG, N).value(FAST, N).$$

We create a theory D_{comb} to describe this specific patient encounter, where D_{cpg} are the union of D_{cpg}^{DU} and D_{cpg}^{TIA}, which were introduced in Section 4.2. The mitigation algorithm begins by applying theorem proving technique and checking if D_{comb} is consistent. Since the theory is consistent, the algorithm infers that no direct interactions exist. At this stage the mitigation algorithm proceeds to the second phase and checks for the existence of an indirect interaction. It starts with IO^1 by formulating the entailment problem $D_{comb} \models \alpha^1$. Because α^1 is not entailed by D_{comb} (i.e., there exists at least one model where α^1 is not satisfied), there are no indirect interactions present in the theory and the mitigation algorithm uses model finding techniques to find a model for the theory D_{comb}. One such model is found and used to create the combined therapy below (for easier readability we omitted the *prec* predicates):

$$executed(PPI).value(UE, H).executed(SC).$$
$$executed(PCS).$$

According to this personalized combined therapy, the patient should be prescribed a proton-pump inhibitor (*executed(PPI)*) and since the result of the endoscopy (UE) is not known (neither *value(UE, H)* nor *value(UE, NH)* is included in D_{pi}), the combined therapy assumes a healed ulcer (*value(UE, H)*) and suggests self-care (*executed(SC)*) for DU and a referral to a primary care specialist for TIA (*executed(PCS)*). Such a combined therapy is returned by the mitigation algorithm and presented to the physician along with the known patient state (D_{pi}). The physician evaluates the therapy by checking the appropriateness of assumptions

diagnosed(TIA)
decision(HG). decision(FAST). decision(NS). decision(RST).
action(EC). action(A). action(TST). action(PCS). action(D). action(NC).
dosage(A, 300). dosage(D, 75).
directPrec(HG, FAST). directPrec(HG, EC). directPrec(FAST, PCS). directPrec(FAST, NS).
directPrec(NS, A). directPrec(NS, TST). directPrec(A, RST). directPrec(TST, NC). directPrec(RST, PCS).
directPrec(RST, D). directPrec(D, NC).

$(value(HG, N) \wedge value(FAST, N) \wedge executed(PCS) \wedge \neg executed(A) \wedge \neg executed(D) \wedge \neg executed(EC) \wedge \neg executed(TST) \wedge \neg executed(NC))$
$\vee$
$(value(HG, N) \wedge value(FAST, P) \wedge value(NS, R) \wedge value(RST, NG) \wedge executed(A) \wedge executed(PCS) \wedge \neg executed(EC) \wedge \neg executed(D) \wedge \neg executed(NC) \wedge \neg executed(TST))$
$\vee$
$(value(HG, N) \wedge value(FAST, P) \wedge value(NS, R) \wedge value(RST, EL) \wedge executed(A) \wedge executed(D) \wedge executed(NC) \wedge \neg executed(EC) \wedge \neg executed(PCS) \wedge \neg executed(TST))$
$\vee$
$(value(HG, N) \wedge value(FAST, P) \wedge value(NS, NR) \wedge executed(TST) \wedge executed(NC) \wedge \neg executed(A) \wedge \neg executed(D) \wedge \neg executed(EC) \wedge \neg executed(PCS))$
$\vee$
$(value(HG, P) \wedge executed(EC) \wedge \neg executed(A) \wedge \neg executed(D) \wedge \neg executed(PCS) \wedge \neg executed(NC) \wedge \neg executed(TST))$

Figure 3: The D_{cpg}^{TIA} theory representing the CPG for TIA.

made, such as the assumption of a healed ulcer in this particular scenario. If she deems some of these assumptions to be inappropriate, new patient information needs to be collected (D_{pi} is updated) and the mitigation algorithm needs to be invoked again to generate a new combined therapy.

4.5 Scenario 2: Adverse Interactions Present

In this scenario we consider a patient suffering from DU who has tested negative for H.pylori (HP) and who on presentation to the emergency department with TIA symptoms has tested negative for hypoglycemia (HG), passed FAST test, and has had neurological symptoms (NS) resolved. The theory D_{pi} describing this patient is:

$diagnosed(DU).value(HP, N).diagnosed(TIA).$
$value(HG, N).value(FAST, P).value(NS, R).$

Similar to the previous scenario, D_{comb} is consistent and as such no direct interactions exist. To check for the existence of an indirect interaction we consider IO^1 and formulate the entailment problem $D_{comb} \models \alpha^1$. This time α^1 is entailed by D_{comb} (it is satisfied by each model of D_{comb}) indicating that an indirect interaction exists.

Following the steps of the mitigation algorithm, we resolve an indirect interaction by selecting a relevant revision operator to revise D_{cpg}. A relevant operator is found by iterating over available revision operators and formulating the entailment problem $D_{comb} \models \beta^k$ for each revision operator RO^k. In this scenario, for RO^1 β^1 is not entailed by D_{comb} as there exists at least one model that does not satisfy β^1. This indicates that RO^1 is not a relevant revision operator. Next, the algorithm considers RO^2 and formulates the entailment problem $D_{comb} \models \beta^2$. β^2 is entailed by D_{comb} and RO^2 is considered a relevant revision operator.

The algorithm revises D_{comb} by modifying D_{cpg} according to the operations Op^2 defined in RO^2. These operations add a proton pump inhibitor (*executed(PPI)*) and reduce the dosage of aspirin by 50 mg to 250 mg (replacing *dosage(A, 300)* with *dosage(A, 250)*). After making these revisions, the mitigation algorithm checks if the revised D_{comb} is consistent. Since it is, the algorithm finds a model for the revised D_{comb} that includes the modified D_{cpg}. This model is used to derive the personalized combined therapy where again the *prec* predicates are excluded for brevity and underlined entries have been introduced by the revision operator:

$value(ZES, N).\underline{executed(PPI)}.value(UE, NH).$
$executed(RS).\underline{executed(A)}.\underline{dosage(A, 250)}.$
$value(RST, EL).executed(\underline{D}).\underline{dosage(D, 75)}.$
$executed(NC).$

According to the combined therapy, the patient is prescribed PPI (*executed(PPI)*) and referred to a specialist for DU (*executed(RS)*) assuming the absence of Zollinger-Ellison syndrome (*value(ZES, N)*) and a not healed ulcer (*value(UE, NH)*). Also the therapy prescribes aspirin (*executed(A)*) with the dosage adjusted to 250mg (*dosage(A, 250)*), prescribes dipyridamole (*executed(D)*) with the dosage set to 75mg (*dosage(D, 75)*), schedules an outpatient neurological consult for TIA (*executed(NC)*)

while at the same time assuming a suspected elevated risk of stroke (*value(RST, EL)*). As in the previous scenario, this combined therapy is presented to the physician for evaluation who may invoke the algorithm again once additional patient information becomes available.

5 Conclusion

Following the recent review of computer-interpretable CPGs (Peleg 2013), our research can be categorized as formal CPG verification. Most of this research involves verifying individual CPGs before they are applied to a patient. For example, Duftschmid and Miksch proposed a knowledge-based detection method for checking the consistency of a CPG represented in ASBRU (Duftschmid and Miksch 2001). Another approach by Perez and Porres uses model checking techniques for authoring and verification of CPGs given in UML (Perez and Porres 2010). Theorem proving techniques were also used to check whether a guideline for managing jaundice in newborns complies with certain properties (ten Teije et al. 2006). Most of the research on identifying adverse interactions in multiple CPGs and mitigating them for comorbid patients relies on description logic for manipulating the guidelines (Jafarpour and Abidi 2013; Abidi et al. 2012; GLI). We believe FOL allows for a more flexible representation by including predicates to represent properties of domain objects and temporal relationships, and flexibly quantified sentences.

In this paper we presented how combining different FOL theories allows us to augment the expressiveness of representation in order to capture intrinsic characteristics of the CPGs and combined therapies, and thus provides for a more complete mitigation framework. Using a simple clinical example we demonstrated the semantic interpretability of the models and combined therapies. In our earlier CLP-based framework we manually interpreted the solutions, distinguishing between action and decision steps, and constructed temporal relationships to impose order on the clinical steps to be taken. The new framework discussed here addresses all of these shortcomings.

For future research, we are working on a different representation of paths in $D^{d_i}_{cpg}$, so disjunctions of conjunctions can be avoided, and on more sophisticated search methods employed by the mitigation algorithm to identify suitable revision operators. Considering that the ultimate goal of our research is to develop a generalized framework of mitigation, we are also studying different clinical situations involving comorbid patients to extract the full set of properties of CPGs that hold across mitigation scenarios.

Acknowledgements: This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada (Collaborative Health Research Program), Telfer School of Management Research Support Program, and the Polish National Science Centre. This research was conducted when Dr. Tan was a postdoctoral fellow with MET Research Group at the University of Ottawa.

References

Abidi, S.; Cox, J.; Abidi, S.; and Shepherd, M. 2012. Using owl ontologies for clinical guidelines based comorbid decision support. In *Proceedings of the 45th Hawaii International Conference on System Science (HICSS 2012)*, 3030–3038.

Duftschmid, G., and Miksch, S. 2001. Knowledge-based verification of clinical guidelines by detection of anomalies. *Artif Intell Med* 22(1):23–41.

Fox, J.; Glasspool, D.; Patkar, V.; Austin, M.; Black, L.; South, M.; Robertson, D.; and Vincent, C. 2010. Delivering clinical decision support services: there is nothing as practical as a good theory. *J Biomed Inform* 43:831–43.

GLINDA: GuideLine INteraction Detection Architecture. glinda-project.stanford.edu/.

Jafarpour, B., and Abidi, S. 2013. Merging disease-specific clinical guidelines to handle comorbidities in a clinical decision support setting. In *Proceedings of the 14th Conference on Artificial Intelligence in Medicine (AIME 2013)*, 28–32.

McCune, W. 2005. Prover9 and mace4. www.cs.unm.edu/ mccune/prover9/.

Michalowski, M.; Wilk, S.; Lin, D.; Michalowski, W.; Tan, X.; and Mohapatra, S. 2013a. Procedural approach to mitigating concurrently applied clinical practice guidelines. In *Proceedings of the First Workshop on Expanding the Boundaries of Health Informatics Using Artificial Intelligence (HIAI13)*.

Michalowski, M.; Wilk, S.; Michalowski, W.; Lin, D.; Farion, K.; and Mohapatra, S. 2013b. Using constraint logic programming to implement iterative actions and numerical measures during mitigation of concurrently applied clinical practice guidelines. In *Proceedings of the 14th Conference on Artificial Intelligence in Medicine (AIME 2013)*, 17–22.

2012. National institute for health and clinical excellence. www.nice.org.uk/.

Pavlov, V.; Schukin, A.; and Cherkasova, T. 2013. Exploring automated reasoning in first-order logic: Tools, techniques and application areas. In *Proceedings of the 4th International Conference on Knowledge Engineering and the Semantic Web (KESW 2013)*, 102–116.

Peleg, M. 2013. Computer-interpretable clinical guidelines: a methodological review. *J Biomed Inform* 46(4):744–763.

Perez, B., and Porres, I. 2010. Authoring and verification of clinical guidelines: a model driven approach. *J Biomed Inform* 43(4):520–536.

Rosenfeld, R., and Shiffman, R. 2009. Clinical practice guideline development manual: a quality-driven approach for translating evidence into action. *Otolaryngol Head Neck Surg* 140(6):1–43.

Sittig, D.; Wright, A.; Osheroff, J.; Middleton, B.; Teich, J.; Ash, J.; Campbell, E.; and Bates, D. 2008. Grand challenges in clinical decision support. *J Biomed Inform* 41:387–92.

ten Teije, A.; Marcos, M.; Balser, M.; van Croonenborg, J.; Duelli, C.; van Harmelen, F.; Lucas, P.; Miksch, S.; Reif, W.; Rosenbrand, K.; and Seyfang, A. 2006. Improving medical protocols by formal methods. *Artif Intell Med* 36(3):193–209.

Wilk, S.; Michalowski, W.; Michalowski, M.; Farion, K.; Hing, M.; and Mohapatra, S. 2013. Automatic mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming. *J Biomed Inform* 46(2):341–353.

Zhang, H., and Zhang, J. 2013. A comparison of finite model generators. *Automated Reasoning and Mathematics* 101–130.