

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

Szymon Wilk¹, Martin Michalowski², Xing Tan³, and Wojtek Michalowski³

¹ Institute of Computing Science, Poznan University of Technology, Poznan, Poland

² Adventium Labs, Minneapolis, USA

³ Telfer School of Management, University of Ottawa, Ottawa, Canada

Abstract. Clinical practice guidelines (CPGs) were originally designed to help with evidence-based management of a single disease and such 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 application 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 [16]. 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 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) [17]. As such, it is believed that executing multiple CPGs concurrently requires a new, “combinatorial, logical, or semantic” methodological approach [2].

Our previous research [19, 8, 7] 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 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 clinical perspective.

This paper is organized as follows. We start with a brief review of related work. Then, 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 provide conclusions and directions for our future research.

2 Related Work

Peleg in her recent methodological review [12] divided the research on computer-interpretable CPGs into eight themes: (1) modeling, (2) acquisition and specification, (3) integration in combination with electronic patient record, (4) validation

and verification, (5) execution, (6) exception handling, (7) maintenance, and finally (8) sharing. According to such categorization, our research discussed here belongs to the validation and verification theme. This theme is further subdivided into three problems: (1) checking for internal consistency and existence of anomalies, (2) checking for existence of desired properties and (3) checking for the inconsistencies between multiple CPGs applied to a comorbid patient.

The first problem from the above list was addressed for example in [1], where the authors proposed a knowledge-based detection method for checking the consistency of a CPG represented in ASBRU. The second problem was described in [14], where model checking was applied for authoring and verification of CPGs represented in UML. Moreover, in [18] theorem proving techniques were used to check whether a guideline for managing jaundice in newborns complies with certain properties. Finally, [13] described a comprehensive framework employing ontological domain knowledge and abductive reasoning to evaluate the completeness and appropriateness of a CPG, and to assess the compliance of physician's actions with this CPG.

The research related to the last verification problem is still in its relatively early stages despite its clinical importance. Proposed solutions vary from manual interventions, where human experts verify and combine multiple CPGs using a specialized editing tool [15], through semi-automatic approaches, where experts resolve automatically discovered conflicts [3], to fully automatic techniques [4, 5]. In [4] the authors proposed an approach that operates on ontological models of CPGs and applies ontology merging techniques to combine these models so that medical, work-flow, institutional and temporal constraints are satisfied. A different approach was described in [5], where individual CPGs are merged according to the combination rules that capture possible drug-drug interactions and prescribe ways of avoiding them.

3 Background

3.1 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* symbols 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* \mathcal{D} is a collection of sentences.

An *interpretation* \mathcal{I} (sometimes called a structure) in FOL is defined as triple:

$$\mathcal{I} = \langle \mathcal{I}_{domain}, \mathcal{I}_{predicate}, \mathcal{I}_{function} \rangle, \quad (1)$$

where

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

Mappings from $\mathcal{I}_{predicate}$ assign meaning to the predicate symbols as follows: for every predicate symbol P of arity n , $\mathcal{I}[P] \in \mathcal{I}_{predicate}$ is an n -ary relation over \mathcal{I}_{domain} , that is $\mathcal{I}[P] \subseteq \mathcal{I}_{domain} \times \dots \times \mathcal{I}_{domain}$.

Mappings from $\mathcal{I}_{function}$ assign meaning to the function symbols as follows: for every function symbol F of arity n , $\mathcal{I}[F] \in \mathcal{I}_{function}$ is an n -ary function over \mathcal{I}_{domain} , that is $\mathcal{I}[F] \in [\mathcal{I}_{domain} \times \dots \times \mathcal{I}_{domain} \rightarrow \mathcal{I}_{domain}]$.

Given an interpretation \mathcal{I} , we can check which sentences of a FOL theory \mathcal{D} are true and which are false according to this interpretation. If sentence $\phi \in \mathcal{D}$ is true given \mathcal{I} , then we write it formally as $\mathcal{I} \models_m \phi$. Moreover, if \mathcal{I} satisfies all sentences in \mathcal{D} , then it is called a *model* for theory \mathcal{D} and formally it is denoted as $\mathcal{I} \models_m \mathcal{D}$.

3.2 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 \mathcal{D} is *consistent* (or satisfiable), iff there exists at least one model of this theory. The question on the consistency of \mathcal{D} can be answered using *theorem proving* [11] that employs automatic reasoning (the resolution method) to construct a proof for \mathcal{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 [20], where possible interpretations are generated until a model is found.

The question about logical consequences is translated into checking if a FOL theory \mathcal{D} entails sentence ϕ (or ϕ is a logical consequence of \mathcal{D}). Formally, we say \mathcal{D} *entails* ϕ , written as $\mathcal{D} \models \phi$, iff, for every interpretation \mathcal{I} such that $\mathcal{I} \models_m \mathcal{D}$, we have $\mathcal{I} \models_m \phi$. In other words, we say \mathcal{D} entails ϕ (or ϕ can be deduced from \mathcal{D}), if ϕ is satisfied by all models for \mathcal{D} .

The entailment $\mathcal{D} \models \phi$ can be translated into checking whether a new theory $\mathcal{D} \cup \{\neg\phi\}$ is not consistent. This means that theorem proving techniques can equivalently be used to check for logical entailments of a theory \mathcal{D} .

4 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.

4.1 Vocabulary

Following our previous work, we assume a CPG is represented as an *actionable graph* (AG) [19]. 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 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 *disease(d)*.

Table 1. Defined predicates

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 <u>path</u> from x to y)
$disease(d)$	d is a disease to be managed
$diagnosed(d)$	the patient has been diagnosed with disease d

4.2 Combined Mitigation Theory

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

$$\mathcal{D}_{comb} = \langle \mathcal{D}_{common}, \mathcal{D}_{cpg}, \mathcal{D}_{pi} \rangle, \quad (2)$$

where:

- \mathcal{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 (for brevity we limit the list to the most relevant ones):
 - $\forall x, y \text{ directPrec}(x, y) \Rightarrow \text{prec}(x, y)$ – association between precedence and direct precedence,
 - $\forall x, y, z \text{ prec}(x, y) \wedge \text{prec}(y, z) \Rightarrow \text{prec}(x, z)$ – transitivity of precedence,
 - $\forall x, y, \text{ prec}(x, y) \wedge \text{prec}(y, x) \Rightarrow x = y$ – antisymmetry of precedence,
 - $\forall x \text{ node}(x) \Rightarrow (\text{action}(x) \wedge \neg \text{decision}(x)) \vee (\neg \text{action}(x) \wedge \text{decision}(x))$ – ensures that a node cannot be simultaneously an action and decision node,
 - $\forall x, n \text{ dosage}(x, n) \Rightarrow \text{action}(x)$ – ensures that only an action node can be characterized with medication dosage,
 - $\forall x, v \text{ value}(x, v) \Rightarrow \text{decision}(x)$ – ensures that only a decision node can be characterized by a value,
 - $\forall d \text{ diagnosed}(d) \Rightarrow \text{disease}(d)$ – ensures that the diagnosed disease is the same as the disease to be managed.
- \mathcal{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:

$$\mathcal{D}_{cpg} = \mathcal{D}_{cpg}^{d_1} \cup \mathcal{D}_{cpg}^{d_2} \cup \dots \cup \mathcal{D}_{cpg}^{d_k}, \quad (3)$$

where $\mathcal{D}_{cpg}^{d_i}$ 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 \mathcal{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,

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

4.3 Interaction and Revision Operators

Interaction and revision operators were introduced in our previous research [19]. Here we reformulate them to account for the FOL-based representation and to enhance their capabilities (e.g., a revision operator may specify multiple operations – details provided below). An interaction operator IO^k encodes knowledge

about indirect adverse interactions (usually drug-drug or drug-disease) and formally it is defined as:

$$IO^k = \langle \alpha^k \rangle, \quad (4)$$

where α^k is a sentence (constructed with the vocabulary described in Section 4.1) describing a specific indirect interaction. Checking whether IO^k is applicable to \mathcal{D}_{comb} (or in other words, if the interaction represented by IO^k occurs in \mathcal{D}_{comb}) is an entailment problem $\mathcal{D}_{comb} \models \alpha^k$.

A revision operator encodes knowledge about the revisions that need to be introduced to the theory \mathcal{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:

$$RO^k = \langle \beta^k, Op^k \rangle, \quad (5)$$

where β^k is a logical sentence that defines the applicability of the operator to the theory \mathcal{D}_{cpg} , and Op^k describes the revisions introduced by RO^k . In particular, Op^k is a set of n pairs of formulas $\langle \phi_i^k, \psi_i^k \rangle$ ($i = 1 \dots n$) that define single operations within the operator. As already stated, these operations are applied only to \mathcal{D}_{cpg} , so other components of \mathcal{D}_{comb} are protected from unwanted revisions. For example, \mathcal{D}_{pi} is never modified thus patient information is never inadvertently changed. The pairs of formulas are interpreted as follows (\emptyset indicates an empty formula):

- $\langle \phi_i^k, \emptyset \rangle$ means that ϕ_i^k is removed from any sentence in \mathcal{D}_{cpg} where it appears,
- $\langle \emptyset, \psi_i^k \rangle$ means that ψ_i^k is added as a new sentence to \mathcal{D}_{cpg} ,
- $\langle \phi_i^k, \psi_i^k \rangle$ means that ϕ_i^k is replaced by ψ_i^k in any sentence in \mathcal{D}_{cpg} where it appears.

It is possible to use unbounded variables in ϕ_i^k and ψ_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 \mathcal{D}_{comb} is analogous to checking the applicability of IO^k and translates into the entailment problem $\mathcal{D}_{comb} \models \beta^k$. In case of direct interactions this entailment problem is simplified – details are given in the next section.

4.4 Mitigation Algorithm

The algorithm consists of two phases and it is outlined in Fig. 1. The first phase involves mitigating direct adverse interactions. Their identification translates into checking the consistency of the \mathcal{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 \mathcal{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 \mathcal{D}_{comb} (specifically its \mathcal{D}_{cpg} component) needs to be revised using applicable revision operators.

Since \mathcal{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 use the following procedure. First, we identify actions shared across individual theories (i.e., theories representing single CPGs) in \mathcal{D}_{comb} . Then, for each shared action x_s we check whether execution of this action and its negation are entailed by individual theories (i.e., $\mathcal{D}_{cpg}^{d_i} \models executed(x_s)$ and $\mathcal{D}_{cpg}^{d_j} \models \neg executed(x_s)$). Such entailments indicate inconsistency caused by x_s . Finally, we identify applicable RO^k by solving a simplified entailment problem: $executed(x_s) \models \beta^k$. The algorithm may stop here, reporting a failure to indicate that \mathcal{D}_{comb} is still inconsistent, if it has failed to address the encountered direct interaction.

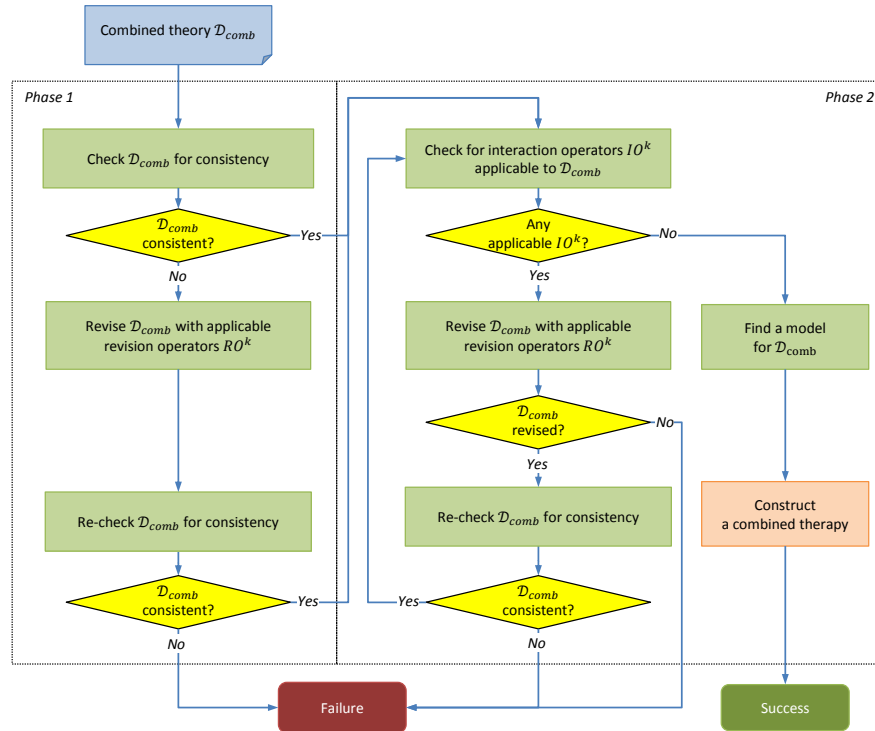


Fig. 1. Outline of the mitigation algorithm.

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 $\mathcal{D}_{comb} \models \alpha^k$). 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 \mathcal{D}_{comb} . This model is equivalent to a solution in the CLP-based mitigation framework, and using its

$\mathcal{I}_{predicate}$ component it is possible to construct a 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* 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 \mathcal{D}_{pi} , thus it is focused on future (suggested) actions and possible (assumed) patient state.

On the other hand, if direct interactions exist (there is at least one IO^k applicable to \mathcal{D}_{comb}), the algorithm attempts to revise \mathcal{D}_{comb} using applicable revision operators, where checking applicability of an operator RO^k is formulated as an entailment problem ($\mathcal{D}_{comb} \models \beta^k$). 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. There is an additional explicit check if \mathcal{D}_{comb} has been revised to avoid indefinite loops if there is no applicable RO^k . If the revised \mathcal{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.

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 [19] 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 [6] that returns a model on top of a theory that has been verified as a consistent one. The performance of Prover9 was verified on a set of benchmark FOL problems and compared to other solvers in [10]. The results show it was among two best performing solvers and for most of the considered problems the proofs were generated in seconds, when running on a personal computer. The running times we observed in our tests were comparable or even shorter, thus they are negligible with regards to the patient management process. Moreover, the mitigation algorithm and its implementation are not bound to Prover9 and Mace4, thus they can be easily replaced by more efficient solvers, if performance becomes an issue.

5 Illustrative Example

In this section we illustrate our proposed FOL-based mitigation framework using the simple clinical case also used in [19]. 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, who 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) [9] and they have been simplified to include only the relevant action and decision nodes.

5.1 Actionable Graphs

Figures 2 and 3 present AGs for DU and TIA simplified guidelines respectively. In these figures the context nodes are indicated with circles, decision nodes are indicated with diamonds, and action nodes with rectangles. The figures also label constants associated with specific nodes and constants corresponding to alternative choices – they are given in square brackets after node and choice descriptions. For example, the HP constant is associated with the “H.pylori” decision node (checking for the presence of helicobacter pylori). There are two alternative choices at this decision node – positive and negative. They are represented as P and N constants respectively.

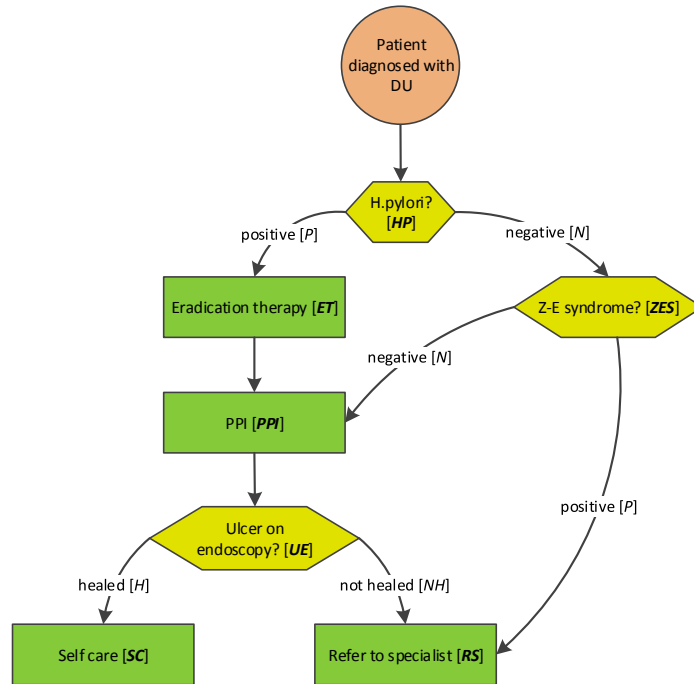


Fig. 2. Actionable graph for DU (AG_{DU}).

5.2 Theories

The AGs are converted into the respective theories, \mathcal{D}_{cpg}^{DU} for DU and \mathcal{D}_{cpg}^{TIA} for TIA, illustrated in Fig. 4 and 5. As can be seen, this representation captures a 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

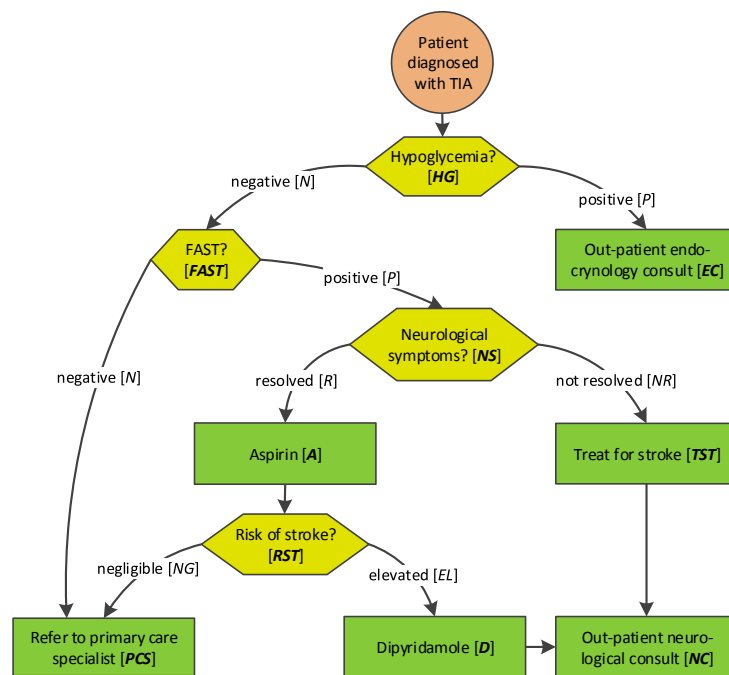


Fig. 3. Actionable graph for TIA (AG_{TIA}).

contains formulas with the negated *executed* predicate to indicate these actions are not executed for a given path.

```

disease(DU).
node(HP). node(ET). node(ZES). node(PPI). node(UE). node(SC). node(RS).
directPrec(HP, ET). directPrec(HP, ZES). directPrec(ET, PPI). directPrec(ZES, PPI).
directPrec(PPI, UE). directPrec(UE, SC). directPrec(UE, RS). directPrec(ZES, RS).
decision(HP). decision(ZES). decision(UE).
action(ET). action(PPI). action(SC). action(RS).
(value(HP, P)  $\wedge$  executed(ET)  $\wedge$  executed(PPI)  $\wedge$  value(UE, H)  $\wedge$  executed(SC)
 $\wedge$   $\neg$ executed(RS))
 $\vee$  (value(HP, P)  $\wedge$  executed(ET)  $\wedge$  executed(PPI)  $\wedge$  value(UE, NH)  $\wedge$  executed(RS)
 $\wedge$   $\neg$ executed(SC))
 $\vee$  (value(HP, N)  $\wedge$  value(ZES, N)  $\wedge$  executed(PPI)  $\wedge$  value(UE, H)  $\wedge$  executed(SC)  $\wedge$ 
 $\neg$ executed(ET)  $\wedge$   $\neg$ executed(RS))
 $\vee$  (value(HP, N)  $\wedge$  value(ZES, N)  $\wedge$  executed(PPI)  $\wedge$  value(UE, NH)  $\wedge$  executed(RS)
 $\wedge$   $\neg$ executed(ET)  $\wedge$   $\neg$ executed(SC))
 $\vee$  (value(HP, N)  $\wedge$  value(ZES, P)  $\wedge$  executed(RS)  $\wedge$ 
 $\neg$ executed(ET)  $\wedge$   $\neg$ executed(PPI)  $\wedge$   $\neg$ executed(SC)).

```

Fig. 4. The D_{cpg}^{DU} theory representing the CPG for DU.

```

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

```

Fig. 5. The D_{cpg}^{TIA} theory representing the CPG for TIA.

5.3 Operators

Interaction and revision operators associated with clinical scenarios discussed below are given in Fig. 6 (for clarity only most relevant operations within revision operators are presented). Their interpretation is as follows:

- 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 is applicable to a patient diagnosed with DU who has been prescribed aspirin (A) without a proton-pump inhibitor (PPI), and has not been prescribed dipyridamole (D). In such case, the patient is taken off of aspirin and prescribed clopidogrel (CL).
- RO^2 is applicable to a patient diagnosed with DU who has been prescribed aspirin (A) without a proton-pump inhibitor (PPI), and also has been prescribed dipyridamole (D). In such case, the patient is prescribed a proton-pump inhibitor (PPI) and dosage of aspirin (A) is reduced by 50 milligrams (mg).

<p>Interaction operators:</p> $IO^1 = \langle \alpha^1 \rangle$ $\alpha^1 = \text{diagnosed}(DU) \wedge \text{executed}(A) \wedge \neg \text{executed}(PPI)$ <p>Revision operators:</p> $RO^1 = \langle \beta^1, Op^1 \rangle$ $\beta^1 = \text{diagnosed}(DU) \wedge \text{executed}(A) \wedge \neg \text{executed}(PPI) \wedge \neg \text{executed}(D)$ $Op^1 = \{ \langle \text{executed}(A), \text{executed}(CL) \rangle \}$ $RO^2 = \langle \beta^2, Op^2 \rangle$ $\beta^2 = \text{diagnosed}(DU) \wedge \text{executed}(A) \wedge \neg \text{executed}(PPI) \wedge \text{executed}(D)$ $Op^2 = \{ \langle \neg \text{executed}(PPI), \text{executed}(PPI) \rangle, \langle \text{dosage}(A, x), \text{dosage}(A, x - 50) \rangle \}$

Fig. 6. Interaction and revision operators.

5.4 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 \mathcal{D}_{pi} describing this patient is given in Fig. 7.

$\text{diagnosed}(DU). \text{value}(HP, P). \text{executed}(ET).$ $\text{diagnosed}(TIA). \text{value}(HG, N). \text{value}(FAST, N).$
--

Fig. 7. The \mathcal{D}_{pi} describing the patient information in Scenario 1.

We create a theory \mathcal{D}_{comb} to describe this specific patient encounter, where \mathcal{D}_{cpg} are the union of \mathcal{D}_{cpg}^{DU} and \mathcal{D}_{cpg}^{TIA} , which were discussed in Section 5.2.

The mitigation algorithm begins by applying theorem proving technique and checking if \mathcal{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 $\mathcal{D}_{comb} \models \alpha^1$. Because α^1 is not entailed by \mathcal{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 \mathcal{D}_{comb} . One such model is found and used to create a combined therapy given in Fig. 8 (for brevity we omitted the *prec* predicates).

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

Fig. 8. Combined therapy created for Scenario 1.

According to the 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 \mathcal{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 (\mathcal{D}_{pi}). The physician evaluates the therapy by checking the appropriateness of assumptions 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 and the mitigation algorithm needs to be invoked again to generate a new combined therapy.

5.5 Scenario 2: Adverse Interactions Present

In this scenario we consider a patient suffering from DU, who has tested negative for H.pylori (HP) and positive for Zollinger-Ellison syndrome (ZES), and who on presentation to the emergency department with TIA symptoms has tested negative for hypoglycemia (HG), passed FAST test, has had neurological symptoms (NS) resolved and for whom the risk of stroke (RST) has been evaluated as elevated. The theory \mathcal{D}_{pi} describing this patient is given in Fig. 9.

diagnosed(DU). value(HP, N). value(ZES, P).
diagnosed(TIA). value(HG, N). value(FAST, P). value(NS, R). value(RST, EL).

Fig. 9. The \mathcal{D}_{pi} describing the patient information in Scenario 2.

Similar to the previous scenario, \mathcal{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 $\mathcal{D}_{comb} \models \alpha^1$. This time α^1 is entailed by \mathcal{D}_{comb} (it is satisfied by each model of \mathcal{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 \mathcal{D}_{cpg} . A relevant operator is found by iterating over available revision operators and formulating the entailment problem $\mathcal{D}_{cpg} \models \beta^k$ for each revision operator RO^k . In this scenario, for RO^1 β^1 is not entailed by \mathcal{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 $\mathcal{D}_{comb} \models \beta^2$. Now β^2 is entailed by \mathcal{D}_{comb} and RO^2 is considered a relevant revision operator.

The algorithm revises \mathcal{D}_{comb} by modifying \mathcal{D}_{cpg} according to the operations Op^2 defined in RO^2 . These operations introduce a proton pump inhibitor (in fact $\neg executed(PPI)$ is replaced by $executed(PPI)$ to avoid direct interaction) 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 \mathcal{D}_{comb} is consistent. Since it is, the algorithm finds a model for the revised \mathcal{D}_{comb} that includes the modified \mathcal{D}_{cpg} . This model is used to derive the combined therapy given in Fig. 10 (again the *prec* predicates are excluded for brevity).

$executed(PPI), executed(RS).$ $executed(A), \underline{dosage(A, 250)}, executed(D), \underline{dosage(D, 75)}, executed(NC).$
--

Fig. 10. Combined therapy created for Scenario 2 (underlined entries have been introduced by the revision operator).

According to the combined therapy in Fig. 10, the patient is prescribed PPI ($executed(PPI)$) and referred to a specialist for DU ($executed(RS)$). Also the therapy prescribes aspirin ($executed(A)$) with the dosage adjusted to 250 mg ($dosage(A, 250)$), prescribes dipyridamole ($executed(D)$) with the dosage set to 75 mg ($dosage(D, 75)$), and schedules an outpatient neurological consult for TIA ($executed(NC)$). As in the previous scenario, such combined therapy is presented for evaluation to the physician who may invoke the algorithm again once additional patient information becomes available.

6 Conclusions

We believe that 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 using 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 had to manually interpret the solutions, distinguishing between action and decision steps, and constructing temporal relationships to impose order in which steps should be taken. The new framework discussed here addresses all these shortcomings.

Presented new framework allows us to deal with such “hard” issues associated with CPGs as, for example, loops. This improved expressiveness comes at the cost of limited comprehensibility by non-experts. However, considering that we envisage the proposed framework to be embedded within a larger clinical decision support system that will present results of mitigation in a user-friendly way, a modeling complexity should not be an issue because actual model will not be seen/presented to a clinician. Only development of the operators will require direct involvement of a clinician, and this process will be guided by a knowledge transfer specialist.

For future research, we are working on a different representation of paths in $\mathcal{D}_{cpg}^{d_i}$, 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.

Acknowledgment

The last two authors were supported by grants from the Natural Sciences and Engineering Research Council of Canada (Collaborative Health Research Program) and Telfer School of Management Research Support Program.

This research was conducted when Dr. Tan was a postdoctoral fellow with MET Research Group at the University of Ottawa.

References

1. Duftschmid, G., Miksch, S.: Knowledge-based verification of clinical guidelines by detection of anomalies. *Artif. Intell. Med.* 22(1), 23–41 (2001)
2. Fox, J., Glasspool, D., Patkar, V., Austin, M., Black, L., South, M., Robertson, D., Vincent, C.: Delivering clinical decision support services: there is nothing as practical as a good theory. *J. Biomed. Inform.* 43, 831–43 (2010)
3. Isern, D., Moreno, A., Pedone, G., Sánchez, D., Varga, L.: Home care personalisation with individual intervention plans. In: Riaño, D. (ed.) *Knowledge Management for Health Care Procedures: ECAI 2008 Workshop, K4HELP 2008*. Revised Selected Papers. pp. 134–151. Springer (2009)
4. Jafarpour, B., Abidi, S.: Merging disease-specific clinical guidelines to handle comorbidities in a clinical decision support setting. In: Peek, N., Morales, R., Peleg, M. (eds.) *14th Conference on Artificial Intelligence in Medicine (AIME 2013)*. Proceedings. pp. 28–32. Springer (2013)

5. López-Vallverdú, J.A., Riaño, D., Collado, A.: Rule-based combination of comorbid treatments for chronic diseases applied to hypertension, diabetes mellitus and heart failure. In: Lenz, R., Miksch, S., Peleg, M., Reichert, M., Riaño, D., ten Teije, A. (eds.) *Process Support and Knowledge Representation in Health Care. BPM 2012 Joint Workshop, ProHealth 2012/KR4HC 2012. Revised Selected Papers.* pp. 30–41. Springer (2013)
6. McCune, W.: *Prover9 and Mace4.* <http://www.cs.unm.edu/~mccune/prover9> (2005)
7. Michalowski, M., Wilk, S., Lin, D., Michalowski, W., Tan, X., Mohapatra, S.: 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)* (2013)
8. Michalowski, M., Wilk, S., Michalowski, W., Lin, D., Farion, K., Mohapatra, S.: Using constraint logic programming to implement iterative actions and numerical measures during mitigation of concurrently applied clinical practice guidelines. In: Peek, N., Morales, R., Peleg, M. (eds.) *14th Conference on Artificial Intelligence in Medicine (AIME 2013).* Proceedings. pp. 17–22. Springer (2013)
9. National Institute for Health and Clinical Excellence (NICE): *NICE pathways.* <http://pathways.nice.org.uk/> (2014)
10. Otten, J.: leanCoP 2.0 and ileanCoP 1.2: High performance lean theorem proving in classical and intuitionistic logic (system descriptions). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) *4th International Joint Conference on Automated Reasoning (IJCAR2008).* Proceedings. pp. 283–291. Springer (2008)
11. Pavlov, V., Schukin, A., Cherkasova, T.: Exploring automated reasoning in first-order logic: Tools, techniques and application areas. In: Klinov, P., Mouromstev, D. (eds.) *4th International Conference on Knowledge Engineering and the Semantic Web (KESW 2013).* Proceedings. pp. 102–116. Springer (2013)
12. Peleg, M.: Computer-interpretable clinical guidelines: a methodological review. *J. Biomed. Inform.* 46(4), 744–763 (2013)
13. Peleg, M., Tu, S., Leonardi, G., Quaglioni, S., Russo, P., Palladini, G., Merlini, G.: Reasoning with effects of clinical guideline actions using OWL: AL amyloidosis as a case study. In: Riaño, D., ten Teije, A., Miksch, S. (eds.) *3th International Workshop on Knowledge Representation for Health Care (KR4HC'11).* Revised Selected Papers. pp. 65–79. Springer (2011)
14. Perez, B., Porres, I.: Authoring and verification of clinical guidelines: a model driven approach. *J. Biomed. Inform.* 43(4), 520–536 (2010)
15. Riaño, D., Real, F., Lopez-Vallverdu, J.A., Campana, F., Ercolani, S., Mecocci, P., Annicchiarico, R., Caltagirone, C.: An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. *J. of Biomed. Inform.* 45(3), 429–446 (2012)
16. Rosenfeld, R., Shiffman, R.: *Clinical practice guideline development manual: a quality-driven approach for translating evidence into action.* *Otolaryngol. Head Neck Surg.* 140(6), 1–43 (2009)
17. Sittig, D., Wright, A., Osheroff, J., Middleton, B., Teich, J., Ash, J., Campbell, E., Bates, D.: Grand challenges in clinical decision support. *J. Biomed. Inform.* 41, 387–92 (2008)
18. ten Teije, A., Marcos, M., Balsemer, M., van Croonenborg, J., Duelli, C., van Harmelen, F., Lucas, P., Miksch, S., Reif, W., Rosenbrand, K., Seyfang, A.: Improving medical protocols by formal methods. *Artif. Intell. Med.* 36(3), 193–209 (2006)

19. Wilk, S., Michalowski, W., Michalowski, M., Farion, K., Hing, M., Mohapatra, S.: Automatic mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming. *J. Biomed. Inform.* 46(2), 341–353 (2013)
20. Zhang, H., Zhang, J.: A comparison of finite model generators. In: Bonacina, M.P., Stickel, M.E. (eds.) *Automated Reasoning and Mathematics. Essays in Memory of William W. McCune.* pp. 101–130. Springer (2013)