MEDINFO 2013
C.U. Lehmann et al. (Eds.)

doi:10.3233/978-1-61499-289-9-352

Clinical Practice Guidelines and Comorbid Diseases: A MiniZinc Representation of Guideline Models for Mitigating Adverse Interactions

Szymon Wilk[a], Martin Michalowski[b], Wojtek Michalowski[c], Ken Farion[d], Di Lin[e], Marisela Mainegra Hing[c], Subhra Mohapatra[f]

[a] *Faculty of Computing Science, Poznan University of Technology, Poland*
[b] *Adventium Labs, Minneapolis, MN, USA*
[c] *Telfer School of Management, University of Ottawa, Ottawa, ON, Canada*
[d] *Division of Emergency Medicine, Children's Hospital of Eastern Ontario, Ottawa, ON, Canada*
[e] *Department of Electrical and Computer Engineering, McGill University, Montreal, PQ, Canada, Canada*
[f] *Faculty of Medicine, University of Ottawa, Ottawa, ON, Canada*

Abstract

Managing a patient with comorbid diseases according to multiple clinical practice guidelines (CPGs) may result in adverse interactions that need to be mitigated (identified and addressed) so a safe therapy can be devised. However, mitigation poses both clinical and methodological challenges. It requires extensive domain knowledge and calls for advanced CPG models and efficient algorithms to process them.

We respond to the above challenges by describing our algorithm that mitigates interactions between pairs of CPGs. The algorithm creates logical models of analyzed CPGs and uses constraint logic programming (CLP) together with domain knowledge, codified as interaction and revision operators, to process them. Logical CPG models are transformed into CLP-CPG models that are solved to find a safe therapy. We represent these CLP-CPG models using MiniZinc, a standard language for CLP models.

As motivation and illustration of our mitigation algorithm we use a clinical case study describing a patient managed for hypertension and deep vein thrombosis according to two individual CPGs. We apply the algorithm to this scenario and present MiniZinc representations of the constructed CLP-CPG models.

Keywords:

Clinical Practice Guideline, Comorbidity, Adverse Drug Event, Constraint Logic Programming, MiniZinc Language.

Introduction

Boyd *et al.* [1] analyzed several hypothetical clinical scenarios of common chronic conditions and associated clinical practice guidelines (CPGs) and concluded that adhering to CPG-mandated therapies in the presence of comorbidities may have undesirable effects on patients. They attributed this situation mostly to the fact that CPGs are developed by a single specialty-dominated committee and therefore do not provide sufficient information for managing comorbid patients. Developing CPGs that explicitly address all potential comorbidities is difficult and impractical therefore there is a need for methods that facilitate the concurrent application of multiple practice guidelines [2]. In fact this is one of the grand challenges for clinical decision support [3].

In this paper we respond to this challenge by describing an algorithm that identifies and addresses adverse interactions between pairs of CPGs (we refer to this process as *mitigation*). This algorithm creates logical models of individual CPGs as well as of their combinations, and employs constraint logic programming (CLP) [4] and domain knowledge codified as *interaction* and *revision operators* to solve them. The former characterize possible indirect adverse interactions, while the latter describe revisions to the logical CPG models that can be applied to address a specific adverse interaction.

A solution to a combined logical model represents a safe therapy (i.e., a therapy with no adverse interactions) that can be applied to a patient, while the lack of a solution indicates adverse interactions (drug-drug or drug-disease) between the CPGs. In cases where no solution exists, the algorithm applies domain knowledge to modify the logical models of the CPGs in order to address the interactions and find a safe therapy.

Our proposed research is defined in a situational context where a patient with two comorbid diseases is managed according to individual, disease specific CPGs. The research builds on our earlier work [5], where we proposed logical models of CPGs and employed the CLP paradigm to evaluate the CPGs for possible interactions. The goal of this study is to demonstrate the ability of the MiniZinc constraint modeling language to represent CLP models derived from the logical CPG models (we refer to them as CLP-CPG models). It facilitates the implementation of the mitigation algorithm, thus we consider this research as a first, yet important, step towards developing a clinical system for the automatic mitigation of multiple CPGs that are applied to manage a patient with comorbid diseases.

Motivation

In this section we describe a clinical case study, developed with assistance from medical experts on our team (physicians with more than 10 years of clinical experience in emergency medicine), that illustrates the motivation behind our research. The scenario considers a patient treated in the emergency department for chronic hypertension (HTN) who is also diagnosed with an acute case of deep vein thrombosis (DVT).

Chronic HTN (blood pressure above 140/90) is a major risk factor for premature cardiovascular morbidity and mortality. Treatment of HTN involves anti-hypertensive drugs along with life style modifications. Patients with a history of HTN may suffer a hypertensive crisis, defined as severe elevation in blood pressure (above 210/120) [6]. Hypertensive crisis is further classified as hypertensive *urgency* or hypertensive *emergency* when associated with end-organ damage. While hypertensive urgency does not constitute a medical emergency, hypertensive emergency must be treated immediately.

DVT is a condition, where a blood clot forms in a deep vein (predominately in the legs). The condition causes a blockage of the blood flow resulting in swelling and pain. The common treatment for DVT is anticoagulation therapy (low molecular weight heparin, or unfractionated heparin in case where additional risk factors such as active ulcers, kidney disease or liver disease are present). In situations where heparins are contraindicated by a history of bleeding tendency, other interventions such as the placement of an inferior vena cava (IVC) filter can be used.

Anticoagulants should not be used when the patient is suffering from hypertensive urgency because of the danger of intracranial hemorrhage. When such an adverse interaction arises, an IVC filter should be used instead of coagulants. However, this interaction is not explicitly described in CPGs for DVT – the physician managing the patient needs to be aware of it and has to modify the therapy accordingly. In order to facilitate the identification of possible adverse interactions and therefore help revising the therapy, a CPG for DVT should be considered alongside a CPG for HTN.

Methods

In this section we present the constructs used by our proposed mitigation algorithm that helps identify and address adverse interactions, such as those associated with the concurrent application of CPGs for HTN and DVT.

Constraint Logic Programming

CLP unifies logic programming (LP) and a constraint satisfaction problem (CSP) by using LP as a constraint programming language to solve a CSP [4]. A logic program is seen as logical theory comprised of a set of rules called clauses. CLP extends this theory by including constraints in the body of the clauses. It queries the program about the provability of a goal to produce a solution to the CLP, where the proof of the goal is composed of clauses whose bodies are satisfiable constraints.

A CLP model is made up of a set of variables, and a set of clauses with constraints and a goal to be satisfied. The clauses in the model capture the relationships between value assignments for variables and they restrict the possible combinations of these assignments. Solving a CLP model entails satisfying the goal given the set of constraints, where a value is assigned to each variable such that no constraints are violated. Variables representing known information are instantiated prior to solving the model and cannot be revised by the solving procedure.

MiniZinc Language

Many constraint satisfaction problems can be solved by CSP solvers using various finite domain and linear programming techniques. However, these solvers use different, often incompatible modeling languages that express problems at varying levels of abstraction. MiniZinc is a medium-level constraint modeling language that has been widely accepted as a standard for CLP models [7]. It can express CLP problems in a solver-independent way and can be easily mapped to various solvers. It also supports different variable types (integers, floats, Booleans) that are accepted by most existing CLP solvers. The characteristics of simplicity, expressiveness, and compatibility with other solvers make MiniZinc a good choice for a standard language and the reason why it was used here to represent CLP-CPG models.

Mitigation Algorithm

We consider two types of adverse interactions — direct and indirect — that may occur, when a patient with comorbid diseases is managed according to simultaneously applied CPGs. Direct adverse interactions are contradictory recommendations given by individual CPGs (e.g., “prescribe aspirin” and “do not prescribe aspirin”), whereas indirect adverse interactions involve drug-drug or drug-disease interactions (e.g., “aspirin increases chances of bleeding and should not be prescribed to patients with an ulcer”) that are not explicitly included in CPGs.

Our mitigation algorithm checks for these direct and indirect adverse interactions between pairs of CPGs for a given patient. If any interactions are identified, the algorithm addresses them by making necessary revisions (e.g., discarding or replacing problematic recommendations) to logical CPG models. Finally, if the interactions are successfully mitigated, the algorithm finds a combined therapy comprised of individual therapies derived from individual CPGs.

Identifying and addressing adverse interactions requires clinical acumen that comes from clinical experts, textbooks, clinical evidence repositories or centralized interaction repositories. The proposed mitigation algorithm assumes that the required domain knowledge is codified in the form of interaction and revision operators.

To make the algorithm independent from any specific CPG representation, we have introduced the concept of an *actionable graph* (AG) to represent the guidelines. An AG can be derived from any CPG representation that distinguishes between context, decision and action steps (according to [8], these steps are common for most CPG representations). Formally, AG is a directed graph with context, action and decision nodes corresponding to context, action and decision steps in a CPG, with arcs corresponding to transitions between the nodes.

The overall structure of the mitigation algorithm is given in Figure 1. The algorithm accepts as input two CPGs represented as AGs and available (possibly incomplete) data that characterizes the current state of the patient. Its three main phases are briefly described in the following sections.

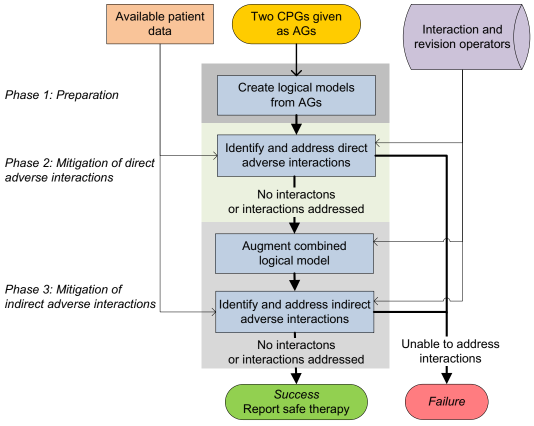

Available patient data
Two CPGs given as AGs
Interaction and revision operators
Phase 1: Preparation
Create logical models from AGs
Phase 2: Mitigation of direct adverse interactions
Identify and address direct adverse interactions
No interactons or interactions addressed
Augment combined logical model
Phase 3: Mitigation of indirect adverse interactions
Identify and address indirect adverse interactions
No interactons or interactions addressed
Unable to address interactions
Success Report safe therapy
Failure

Figure 1 – Overall structure of the mitigation algorithm

Phase 1: Preparation

This phase starts by constructing *logical models* that are logical representations of AGs (and thus the underlying CPGs). Formally, a logical model of the actionable graph AG_i is defined as $LM_i = <d_i, V_i, PLE_i>$, where d_i is the label of a disease (defined by the context node) associated with AG_i, V_i is a set of variables associated with actions and decision nodes in AG_i, and PLE_i is a set of logical expressions representing paths in AG_i, each being a conjunction of variable-value pairs (the process of translating paths into logical expressions is described in our previous work [9]). All action variables in LM_i are Boolean where *true* indicates an action associated with a variable should be carried out, while *false* indicates it should not. Decision variables have domains defined by possible choices and assigning a specific value to a decision variable identifies the choice that is made.

Logical models are brought together as a *combined logical model* $CLM_{i,j}$ that is formally defined as: $CLM_{i,j} = \langle LM_i, LM_j, ILE_{i,j} \rangle$, where $ILE_{i,j}$ is a set of logical expressions that represent indirect adverse interactions between AG_i and AG_j. Initially $ILE_{i,ji}$ is empty and the algorithm updates it in the third phase.

Phase 2: Mitigation of Direct Adverse Interactions

In the second phase, the algorithm checks for direct adverse interactions associated with variables shared by logical models LM_i and LM_j (if there are no shared variables, the algorithm will skip this phase). It first transforms the $CLM_{i,j}$ into a CLP-CPG model represented in MiniZinc. A MinZinc model includes variables from V_i and V_j, one constraint for PLE_i and one for PLE_j, each represented as a disjunction of conjunctions (one conjunction per logical expression), and a set of constraints corresponding to negated logical expressions from $ILE_{i,j}$ (these constraints ensure indirect interactions are avoided and are introduced only in MiniZinc models created in the third stage of the algorithm, thus no constraints from $ILE_{i,j}$ are introduced in this stage).

Using a CLP solver (i.e., the MiniZinc MIP solver), the algorithm attempts to solve the CLP-CPG model instantiated with the available patient data. A solution to this model, denoted as $SOL_{i,j}$, is an assignment of values to variables such that all constraints are satisfied. A solution for the CLP-CPG model is also a solution for $CLM_{i,j}$. If such a solution exists, it means no direct interactions exist and the algorithm progresses to the third stage. No solution indicates that some constraints were violated as a result of direct interactions. Variables appearing in the violated constraints form a *potential source of infeasibility* denoted as $PSI_{i,j.}$

$PSI_{i,j}$ drives the process for addressing adverse interactions, where the algorithm applies revision operators to the combined logical model. A revision operator is formally defined as $RO^k = \langle D^k, V^k, sle^k, tle^k \rangle$, where D^k is a set of disease labels to which the operator can be applied (D^k can include the '*' wildcard indicating any disease), V^k is a set of variables and sle^k and tle^k are logical expressions. D^k and V^k form the activation component, and sle^k and tle^k form the knowledge component of the operator.

The activation component indicates that the revision operator RO^k is activated when D^k contains '*' or any of the disease labels associated with the current combined logical model $CLM_{i,j}$, and when V^k is included in the encountered potential source of infeasibility $PSI_{i,j}$. Moreover, the knowledge component of RO^k states that whenever sle^k appears in logical expressions from ILE_i or ILE_j, it is replaced by tle^k.

The algorithm checks if all interactions have been addressed by creating and solving a new CLP-CPG model from the revised combined logical model (if more than one revision operator has been activated, they are considered iteratively and individually). The lack of a solution after exhausting all applicable revision operators signifies direct interactions that cannot be addressed and the algorithm terminates by reporting failure to produce a solution. If a solution is found, the algorithm proceeds to the third phase.

Phase 3: Mitigation of Indirect Adverse Interactions

The third mitigation phase is very similar to the second phase — the main difference is the application of interaction operators to the combined logical model at the very beginning of this phase.

An interaction operator is formally defined as $IO^k = \langle D^k, V^k, le^k \rangle$, where again D^k is a set of disease labels to which the operator can be applied (or the '*' wildcard), V^k is a set of variables and le^k is a logical expression. D^k and V^k form the activation component, while le^k is the knowledge component.

The interaction operator, IO^k, is activated if D^k includes '*' or any of the disease labels from the combined logical model $CLM_{i,j}$, and where V^k is a subset of variables from $CLM_{i,j}$. Once IO^k has been activated, the knowledge component le^k that codifies a single adverse interaction is added to $ILE_{i,j}$. All activated interaction operators are applied to $CLM_{i,j}$ in order to augment it with information about possible drug-drug or drug-disease interactions. The subsequent processing is analogous to the second phase.

Finally, the algorithm reports *failure* if CPGs cannot be applied simultaneously to a patient (due to adverse interactions left unresolved), or *success* if the application of CPGs is possible. In the latter case the algorithm also reports the resulting combined therapy (represented by $SOL_{i,j}$).

Application

In this section the mitigation algorithm is applied using two scenarios derived from the motivating case study presented earlier. CPGs for DVT and HTN represented as AGs are given in Figures 2 and 3 respectively (they have been simplified for better understanding). These figures also label variables associated with specific action and decision nodes (in square brackets next to node labels), as well as possible values of decision variables (in square brackets next to arc labels).

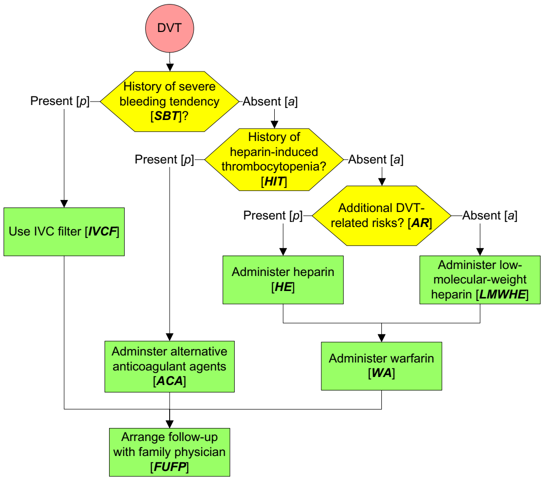

DVT
History of severe bleeding tendency [SBT]?
Present [p]
Absent [a]
History of heparin-induced thrombocytopenia? [HIT]
Present [p]
Absent [a]
Additional DVT-related risks? [AR]
Present [p]
Absent [a]
Use IVC filter [IVCF]
Administer heparin [HE]
Administer low-molecular-weight heparin [LMWHE]
Adminster alternative anticoagulant agents [ACA]
Administer warfarin [WA]
Arrange follow-up with family physician [FUFP]

Figure 2 – CPG for DVT represented as an actionable graph (AG_{DVT})

Figure 4 lists interaction and restriction operators related to these two conditions. The former indicate selected adverse interactions between various anticoagulant agents and hypertensive urgency, while the latter address a possible source of infeasibility by replacing anticoagulants with an IVC filter. In all logical expressions in Figure 4 and in the text we use a simplified notation for Boolean variables – i.e., V = *true* is denoted simply as V, while V = *false* as $\neg V$.

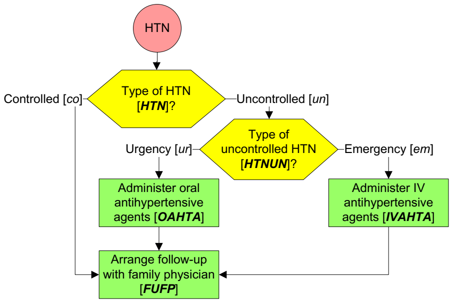

HTN
Type of HTN [*HTN*]?
Controlled [*co*]
Uncontrolled [*un*]
Type of uncontrolled HTN [*HTNUN*]?
Urgency [*ur*]
Emergency [*em*]
Administer oral antihypertensive agents [*OAHTA*]
Administer IV antihypertensive agents [*IVAHTA*]
Arrange follow-up with family physician [*FUFP*]

Figure 3 – CPG for HTN represented as an actionable graph (AG_{HTN})

Interaction operators
$IO^1 := <\{*\}, \{HTNUN, ACA\}, (HTNUN = ur) \wedge ACA>$
$IO^2 := <\{*\}, \{HTNUN, HE, WA\}, (HTNUN = ur) \wedge HE \wedge WA>$
$IO^3 := <\{*\}, \{HTNUN, LMWHE, WA\}, (HTNUN = ur) \wedge LMWHE \wedge WA>$

Revision operators
$RO^1 := <\{*\}, \{HTNUN, ACA\}, (ACA \wedge \neg IVCF), (\neg ACA \wedge IVCF)>$
$RO^2 := <\{*\}, \{HTNUN, HE, WA\},$
$(HE \wedge WA \wedge \neg IVCF), (\neg HE \wedge \neg WA \wedge IVCF)>$
$RO^3 := <\{*\}, \{HTNUN, LMWHE, WA\},$
$(LMWHE \wedge WA \wedge \neg IVCF), (\neg LMWHE \wedge \neg WA \wedge IVCF)>$

Figure 4 – Domain knowledge related to DVT and HTN and codified as operators

Scenario 1: No adverse interactions

We consider a patient experiencing uncontrolled hypertensive emergency with a history of severe bleeding tendency. This implies invoking the algorithm with the following patient data: $HTN := un$, $HTNUN := em$, $SBT := p$.

In phase 1, the mitigation algorithm constructs the logical models presented in Figure 5. Note that expressions in PLE_{DVT} and PLE_{HTN} include negated action variables that do not appear in a particular path. The intuition behind this representation is that only actions explicitly indicated by the path should be taken and others should not. The combined logical model $CLM_{DVT,HTN}$ at this point includes no information about indirect adverse interactions – its $ILE_{DVT,HTN}$ component is empty.

$LM_{DVT} = <d_{DVT}, V_{DVT}, PLE_{DVT}>$
d_{DVT} = DVT
$V_{DVT} = \{SBT, HIT, AR, IVCF, ACA, HE, LMWHE, WA, FUFP\}$
$PLE_{DVT} = \{$
$(SBT = p) \wedge IVCF \wedge FUFP \wedge \neg ACA \wedge \neg HE \wedge \neg LMWHE \wedge \neg WA,$
$(SBT = a) \wedge (HIT = p) \wedge ACA \wedge FUFP \wedge$
$\neg IVCF \wedge \neg HE \wedge \neg LMWHE \wedge \neg WA,$
$(SBT = a) \wedge (HIT = a) \wedge (AR = p) \wedge HE \wedge WA \wedge FUFP \wedge$
$\neg IVCF \wedge \neg ACA \wedge \neg LMWHE,$
$(SBT = a) \wedge (HIT = a) \wedge (AR = a) \wedge LMWHE \wedge WA \wedge FUFP \wedge$
$\neg IVCF \wedge \neg ACA \wedge \neg HE\}$

$LM_{HTN} = <d_{HTN}, V_{HTN}, PLE_{HTN}>$
d_{HTN} = HTN
$V_{HTN} = \{HTN, HTNUN, OAHTA, IVAHTA, FUFP\}$
$PLE_{HTN} = \{$
$(HTN = co) \wedge FUFP \wedge \neg OAHTA \wedge \neg IVAHTA,$
$(HTN = un) \wedge (HTNUN = ur) \wedge OAHTA \wedge FUFP \wedge \neg IVAHTA,$
$(HTN = un) \wedge (HTNUN = em) \wedge IVAHTA \wedge FUFP \wedge \neg OAHTA\}$

$CLM_{DVT,HTN} = <LM_{DVT}, LM_{HTN}, \{\}>$

Figure 5 – Logical models created in phase 1

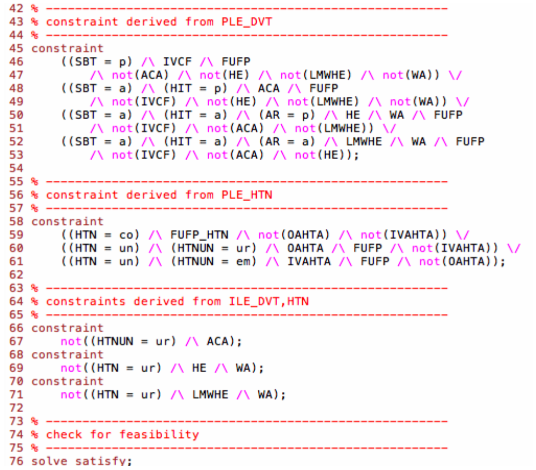
The logical models LM_{DVT} and LM_{HTN} share one variable – *FUFP*, therefore the algorithm proceeds to phase 2 and creates a CLP-CPG model from $CLM_{DVT,HTN}$ (for the sake of brevity we skip these details). The CLP-CPG model in this phase has a solution indicating the lack of direct adverse interactions.

Next, the algorithm enters phase 3 and activates applicable interaction operators. All three operators given in Figure 4 (IO^1, IO^2, IO^3) can be activated (they are applicable to any disease and all variables present in their activation components appear in $CLM_{DVT,HTN}$). These operators augment the combined logical model by adding logical expressions from their knowledge components to $ILE_{DVT,HTN}$. This model is given in Figure 6.

$CLM_{DVT,HTN} = <LM_{DVT}, LM_{HTN}, ILE_{DVT,HTN}>$
$ILE_{DVT,HTN} = \{$
$(HTNUN = ur) \wedge ACA,$
$(HTNUN = ur) \wedge HE \wedge WA,$
$(HTNUN = ur) \wedge LMWHE \wedge WA\}$

Figure 6 – Combined logical model augmented in phase 3

$CLM_{DVT,HTN}$ from Figure 6 is represented in the MiniZinc language (the main parts are shown in Figure 7 – note that expressions from $ILE_{DVT,HTN}$ have been negated) and solved. The algorithm reports *success* together with the following solution (limited to action variables): $IVCF := true$, $IVAHTA := true$, $FUFP := true$, $OAHTA := false$, $ACA := false$, $WA := false$, $HE := false$, $LMWHE := false$. Translated into layman terms the solution represents a combined therapy of implanting the IVC filter (*IVCF*) to manage DVT and giving IV antihypertensive agents (*IVAHTA*) to manage HTN. A follow-up with a family physician (*FUFP*) is recommended.


```
% ------------------------------------------------------------
% constraint derived from PLE_DVT
% ------------------------------------------------------------
constraint
    ((SBT = p) /\ IVCF /\ FUFP
        /\ not(ACA) /\ not(HE) /\ not(LMWHE) /\ not(WA)) \/
    ((SBT = a) /\ (HIT = p) /\ ACA /\ FUFP
        /\ not(IVCF) /\ not(HE) /\ not(LMWHE) /\ not(WA)) \/
    ((SBT = a) /\ (HIT = a) /\ (AR = p) /\ HE /\ WA /\ FUFP
        /\ not(IVCF) /\ not(ACA) /\ not(LMWHE)) \/
    ((SBT = a) /\ (HIT = a) /\ (AR = a) /\ LMWHE /\ WA /\ FUFP
        /\ not(IVCF) /\ not(ACA) /\ not(HE));

% ------------------------------------------------------------
% constraint derived from PLE_HTN
% ------------------------------------------------------------
constraint
    ((HTN = co) /\ FUFP_HTN /\ not(OAHTA) /\ not(IVAHTA)) \/
    ((HTN = un) /\ (HTNUN = ur) /\ OAHTA /\ FUFP /\ not(IVAHTA)) \/
    ((HTN = un) /\ (HTNUN = em) /\ IVAHTA /\ FUFP /\ not(OAHTA));

% ------------------------------------------------------------
% constraints derived from ILE_DVT,HTN
% ------------------------------------------------------------
constraint
    not((HTNUN = ur) /\ ACA);
constraint
    not((HTN = ur) /\ HE /\ WA);
constraint
    not((HTN = ur) /\ LMWHE /\ WA);

% ------------------------------------------------------------
% check for feasibility
% ------------------------------------------------------------
solve satisfy;
```

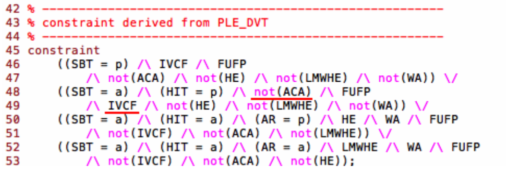
Figure 7 – The MiniZinc representation of $CLM_{DVT,HTN}$

Scenario 2: Indirect adverse interactions

Next we consider a patient with uncontrolled hypertensive urgency, no history of severe bleeding tendency, and a history of heparin-induced thrombocytopenia. Thus, the algorithm is run with the following patient data: $HTN := un$, $HTNUN := ur$, $SBT := a$, $HIT := p$.

The mitigation algorithm completes phases 1 and 2 with the same results as in Scenario 1 (see Figure 5), and the combined logical model augmented with interaction operators and its MiniZinc representation is identical (see Figures 6 and 7 respectively). This time the solver fails to find a solution because the constraint $\neg((HTNUN = ur) \wedge ACA)$ (introduced by the interaction operator IO^1) is violated. This indicates an indirect adverse interaction – due to history of thrombocytopenia ($HIT := p$) the patient should be prescribed alternative anticoagulants (*ACA*), however such a treatment is not appropriate in the presence of hypertensive urgency ($HTNUN = ur$). Variables appearing in the violated constraint define the potential source of infeasibility ($PSI_{DVT,HTN} := \{HTNUN, ACA\}$).

Next, the mitigation algorithm activates the revision operators applicable to $PSI_{DVT,HTN}$. There is one such operator – RO^1 – that replaces alternative anticoagulants (*ACA*) with an IVC filter (*IVCF*), and this operator is applied to $CLM_{DVT,HTN}$. The revised $CLM_{DVT,HTN}$ is used to construct a new CLP-CPG model in MiniZinc. The revised part of the MiniZinc model is given in Figure 8 (revisions are underlined).



```
% --------------------------------------------------------
% constraint derived from PLE_DVT
% --------------------------------------------------------
constraint
    ((SBT = p) /\ IVCF /\ FUFP
        /\ not(ACA) /\ not(HE) /\ not(LMWHE) /\ not(WA)) \/
    ((SBT = a) /\ (HIT = p) /\ not(ACA) /\ FUFP
        /\ IVCF /\ not(HE) /\ not(LMWHE) /\ not(WA)) \/
    ((SBT = a) /\ (HIT = a) /\ (AR = p) /\ HE /\ WA /\ FUFP
        /\ not(IVCF) /\ not(ACA) /\ not(LMWHE)) \/
    ((SBT = a) /\ (HIT = a) /\ (AR = a) /\ LMWHE /\ WA /\ FUFP
        /\ not(IVCF) /\ not(ACA) /\ not(HE));
```

Figure 8 – The MiniZinc representation of the revised combined logical model

Solving the revised $CLM_{DVT,HTN}$ yields a solution as the potential source of infeasibility has been successfully addressed. The algorithm reports success together with the following solution – *IVCF* := *true*, *OAHTA* := *true*, *FUFP* := *true*, *ACA* := *false*, *HE* := *false*, *LMWHE* := *false*, *WA* := *false*, *IVAHTA* := *false*. In other terms, the combined therapy asks for implanting an IVC filter (*IVCF*) to manage DVT, prescribing oral antihypertensive agents (*OVAHTA*) to manage HTN, and recommends a follow-up with a family physician (*FUFP*).

Discussion

Applying CPGs to a patient with comorbid diseases is a challenging problem, both clinically and methodologically. As stated in [10], combining multiple CPGs "is not a trivial task" and this may explain the relatively slow progress of this line of research. Solutions proposed thus far vary from human-driven approaches, where experts combine CPGs using a specialized editing tool [11], semi-automatic approaches where experts resolve conflicts discovered by automatic methods [10], to fully automatic approaches that rely on codified domain knowledge [12]. For example, Real and Riaño [12] describe a method that decomposes CPGs into state-action pairs, expands them into state-action-prognosis triples and combines the triples into a single CPG using restriction and substitution rules. While substitution rules are conceptually similar to the interaction and revision operators, our approach allows for a finer description of interactions. Interaction operators share some similarity with safety rules proposed in Peleg *et al.* [13] that enhance a single CPG with information about possible adverse interactions. However, our operators are independent of specific CPGs and can be used across a number of clinical domains, fostering the sharing of clinical knowledge.

In this paper we took an important step towards the operationalization of guidelines by showing how CLP-CPG models derived from logical CPG models can be represented using the MiniZinc language. An efficient mechanism for solving CLP problems associated with mitigating adverse interactions is crucial to the ultimate goal of our research, namely to include the automatic mitigation algorithm as part of a mobile clinical decision support system for use at the point of care.

The presented algorithm has two limitations. First, we restrict the algorithm to automatically evaluating two CPGs at a time to avoid the exponential increase in the number of possible modifications to the logical models. Second, we assume that the mitigation of CPGs occurs at the action level and does not involve automatic adjustments of dosages of medications. Our current work is focused on expanding the mitigation algorithm to address these two limitations.

Acknowledgments

The research described in this paper was supported by grants from the Natural Sciences and Engineering Research Council of Canada, University of Ottawa, and from the Polish National Science Centre.

References

[1] Boyd CM, Darer J, Boult C, Fried LP, Boult L, and Wu AW. Clinical practice guidelines and quality of care for older patients with multiple comorbid diseases: implications for pay for performance. JAMA 2005: 294 (6): 716-24.

[2] Fox J, Glasspool D, Patkar V, Austin M, Black L, South M, Robertson D, and Vincent C. Delivering clinical decision support services: there is nothing as practical as a good theory. J Biomed Inform 2010: 43 (5): 831-43.

[3] Sittig DF, Wright A, Osheroff JA, Middleton B, Teich JM, Ash JS, Campbell E, and Bates DW. Grand challenges in clinical decision support. J Biomed Inform 2008: 41 (2): 387-92.

[4] Dechter R. Constraint Processing: The MIT Press, 1989.

[5] Wilk S, Michalowski W, Michalowski M, Farion K, Hing MM, and Mohapatra S. Mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming. J Biomed Inform 2013: 42 (2): 341-53.

[6] Epstein M. Diagnosis and management of hypertensive emergencies. Clin Cornerstone 1999: 2 (1): 41-54.

[7] Nethercote N, Stuckey P, Becket R, Brand S, Duck G, and Tack G. MiniZinc: Towards a standard CP modeling language. In: Proc of the 13th International Conference on Principles and Practice of Constraint Programming, CP'07: Springer, 2007; pp. 529-43.

[8] Isern D, and Moreno A. Computer-based execution of clinical guidelines: a review. Int J Med Inform 2008: 77 (12): 787-808.

[9] Wilk S, Michalowski M, Michalowski W, Hing MM, and Farion K. Reconciling pairs of concurrently used clinical practice guidelines using constraint logic programming. AMIA Annu Symp Proc 2011: 944-53.

[10] Isern D, Moreno A, Pedone G, Sánchez D, and Varga L. Home care personalisation with individual intervention plans. In: Knowledge Management for Health Care Procedures, K4HElP 2008: Springer, 2009; pp. 134-51.

[11] Riano D, Real F, Lopez-Vallverdu JA, Campana F, Ercolani S, Mecocci P, Annicchiarico R, and Caltagirone C. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. J Biomed Inform 2012: 45 (3): 429-46.

[12] Real F, and Riaño D. Automatic combination of formal intervention plans using SDA* representation model. In: Knowledge Management for Health Care Procedures, K4CARE 2007: Springer, 2008; pp. 75-86.

[13] Peleg M, Tu SW, Leonardi G, Quaglini S, Russo P, Palladini G, and Merlini G. Reasoning with effects of clinical guideline actions using OWL: AL amyloidois as a case study. In: Knowledge Representation for Health-Care, KR4HC 2011: Springer, 2011; pp. 65-79.

Address for correspondence

Szymon Wilk, Faculty of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland
szymon.wilk@cs.put.poznan.pl