A Constraint Logic Programming Approach to Identifying Inconsistencies in Clinical Practice Guidelines for Patients with Comorbidity

Martin Michalowski[1], Marisela Mainegra Hing[2], Szymon Wilk[3], Wojtek Michalowski[2], and Ken Farion[4]

[1] Adventium Labs, Minneapolis MN 55401, USA,
martin.michalowski@adventiumlabs.org
[2] University of Ottawa, Telfer School of Management, Ottawa Canada,
{wojtek,mhing}@uottawa.ca
[3] Poznan University of Technology, Institute of Computing Science, Poznan Poland,
szymon.wilk@cs.put.poznan.pl
[4] Children Hospital of Eastern Ontario, Ottawa Canada,
farion@cheo.on.ca

Abstract. This paper describes a novel methodological approach to identifying inconsistencies when concurrently using multiple clinical practice guidelines. We discuss how to construct a formal guideline model using Constraint Logic Programming, chosen for its ability to handle relationships between patient information, diagnoses, and treatment suggestions. We present methods to identify inconsistencies that are manifested by treatment-treatment and treatment-disease interactions associated with comorbidity. Using an open source constraint programming system (ECLiPSe), we demonstrate the ability of our approach to find treatment given incomplete patient data and to identify possible inconsistencies.

Key words: Clinical practice guideline, comorbidity, Constraint Logic Programming

1 Introduction

This paper describes methodological research on developing a guideline model based on Constraint Logic Programming (CLP) [1] that allows for the semi-automatic adaptation of a guideline to a patient-specific situation characterized by the need to treat multiple diseases concurrently (*comorbidity*). A clinical practice guideline (CPG) represents systematically developed statements to assist physician decision-making, aiming to improve quality of care and reduce practice variations. CPGs were originally intended to help with managing a patient who has a single medical condition. While the population is aging and medical care involves increasingly complex cases, there is clearly a need to adapt guidelines for management of patients with comorbidity.

The use of CPGs at the point of care is limited and a number of barriers have been identified [2, 3]. One particular barrier is that the CPG seldom allows for

customization to a patient-specific situation, especially in light of comorbidity. This shortcoming can be addressed by creating a computer executable model of the guideline that can be easily manipulated and tailored to a specific patient. In our research we consider simplified intervention CPGs that were developed to assist physicians in one-time disease management and that are used in a primary care organization on the in-patient population. The executable CPG model is build using the constraint logic programming (CLP) paradigm and we developed an approach to identify potential inconsistencies if multiple CPG models need to be evaluated concurrently. This places our work in the broad category of reasoning in medicine [4] and our attempt to deal with inconsistencies shares some similarities with conflict resolution methods in automated reasoning [5].

This paper is organized as follows. First we describe CLP methodology and show how we use this framework in the context of CPGs. Next we describe how we model a single CPG as a CLP and show how a model for comorbid condition is created. We then outline how such a combined model can introduce inconsistencies and present methods for resolving them. We continue by presenting a case study and summarizing our contributions.

2 Methods

CLP is a form of constraint programming that combines logic programming (LP) with a constraint satisfaction problem (CSP) by embedding constraints within the body of clauses in a logic program. Formally, a logic program is seen as logical theory made up of a set of rules called *clauses* that relate the truth value of a literal (an n-ary predicate where the n terms can be either a variable, constant, or a n-ary function) to the value of a collection of other literals. Executing a logic program entails asking for the truth value of a certain statement called the goal. CLP unifies LP and constraint satisfaction problems (CSPs) [1] by using logic programming as a constraint programming language to solve a CSP.

We use CLP over finite domain constraints because the constraints we represent are over binary (true/false). In the CLP representation of a guideline, a variable is associated with an action or decision step from a CPG (i.e., take oral cortiosteriods (OC) or complicated ulcer symptoms experienced (CUS)). Than, an instance of a variable is $OC :=$ true or $CUS :=$ false. A constraint describes the relationship between variables and is represented in the body of a clause in the logic program. For example $\neg OC \wedge CUS$ means a patient cannot take oral cortiosteriods when experiencing ulcer symptoms. Solving the CLP model entails assigning a value to each variable such that no constraints are violated. As such, the solution task can be approached in three different ways: (1) model checking (to determine whether the problem has a solution), (2) search to find a single solution, or (3) search to find all solutions. In this research, we solve the CLP model by searching to find a single solution.

3 Modeling and Combining Multiple CPGs

We start with generating a CLP model for a CPG along with an associated knowledge base ($\mathcal{K}$) containing external information about adverse interactions

between patient's conditions. Second, for a patient with comorbidity all individual models relevant to a patient's condition are combined into a single CLP model and this combined model is solved. A solution includes a set of valid actions and when one doesn't exist, we identify the inconsistencies.

3.1 Modeling a Single CPG

We start with a flowchart CPG representation that is converted to a decision graph from which logical rules are derived (each possible path in the decision graph becomes a rule). These rules are used to generate a CLP model of the guideline. For the sake of brevity we will describe in detail only the last step of this modeling process (details of the former steps are presented in [6]). As a working example, we use a peptic ulcer condition. The starting point for the modeling process is the set of logical rules generated from the AIHA[1] Peptic Ulcer in Adults CPG.

We augment a CPG with information from a knowledge base representing external knowledge not included in the guideline. Drug interactions that are established from the Epocrates database[2] are examples of knowledge that affects the realization of a patient's treatment plan. An external knowledge base is represented as a set of clauses. Given the set of logical rules representing a CPG and a set of clauses representing an external knowledge base associated with the given CPG, we map them into a CLP model. All variables in the CLP model correspond to the predicates in the rules. We refer to this model as a *CLP-CPG model* and one for peptic-ulcer is shown in Figure 1.

$$\mathcal{V} = (CUS, SC, CNSAID, NSAID, PET, PHPT, ET, OD, OC, IC)$$

$$\mathcal{CL} = \left\{ \begin{array}{c} CUS \rightarrow SC \equiv true \\ \neg CUS \wedge CNSAID \rightarrow \neg NSAID \equiv true, \\ \neg CUS \wedge \neg CNSAID \wedge PET \rightarrow SC \equiv true, \\ \neg CUS \wedge \neg CNSAID \wedge \neg PET \wedge PHPT \rightarrow ET \equiv true, \\ \neg CUS \wedge \neg CNSAID \wedge \neg PET \wedge \neg PHPT \rightarrow OD \equiv true, \\ \neg OC \equiv true,\ \neg NSAID \equiv true,\ \neg(ET \wedge IC) \equiv true \end{array} \right\}$$

Fig. 1. CLP-CPG model of the peptic-ulcer CPG

A rule-based model of the CPG always has a feasible assignment of values because the original CPG, by definition, always provides diagnosis and a treatment recommendation when presented with valid patient information. Moreover, the clauses introduced by the external knowledge base do not affect feasibility, thus by transitivity, a CLP-CPG model will also always provide a solution given a valid instantiation of variables. However, when combining CLP-CPG models to create a combined CLP model representing multiple CPGs, we can no longer guarantee the existence of a solution.

[1] www.aiha.org
[2] www.epocrates.com

3.2 Combining Multiple CPGs

In a combined CLP-CPG model, the set of variables is simply the superset of variables in the individual models and the set of constraints is the union of constraints from each model. In cases where there are no shared variables between single models, solving the combined model is equivalent to solving each CLP-CPG independently. In the case of a combined model with variables present across multiple constraints, this dependence results in a solution that is constrained by the interaction of multiple CPGs.

There are several challenges associated with the concurrent use of CPGs and here we focus on those that involve inconsistencies. The inconsistencies can occur at two levels. One is related to drug interactions when combining different medications and the other is related to drug-disease interactions [7] where a certain medication can exacerbate symptoms of a condition. Thus, when combining multiple CLP-CPG models to account for comorbid condition, it is necessary to identify inconsistencies if they exist. We call such inconsistencies *points of contention* (*POC*).

Formally, a POC is defined as follows:

Definition 1. *A POC is a set of variables T in the combined CLP-CPG model whose domains are annihilated (reduced to the empty set) during search, resulting in no found solution.*

A POC occurs when given a patient's state one CPG explicitly identifies a class of actions that are inadmissible according to some actions defined by another CPG. When considering a combined CLP-CPG model, we are also able to identify a POC that is not evident from consulting only the CPGs themselves but becomes apparent when external knowledge is introduced. The first step to eliminating a POC is to check whether it is due to a variable's value assigned during search for a solution or due to a variable with a value predefined according to available patient information. In the first case we use standard backtrack search to find a valid solution representing a course of feasible actions. If backtracking does not produce a solution or the POC is caused by a variable's predefined value, the problem becomes more difficult. Our approach is to present the physician with a partial solution along with the identified POC. The partial solution includes assignments of values to variables not included in the POC and the POC itself provides information that is used by the physician in making treatment decisions. We use backmarking [1], available as part of the ECLiPSe[3] solver, as a method to generate a partial solution.

3.3 A Case Study

To illustrate our proposed approach, we generated CLP-CPG models from asthma and peptic ulcer CPGs using the process described in Section 3.1. We solve the combined CLP-CPG model using the ECLiPSe system. We constructed various real-world scenarios for patients with comorbidity involving asthma exacerbation and peptic ulcer by assigning different initial values to variables. Due to a lack of space, we only summarize our results below.

[3] www.eclipseclp.org

For scenarios with no POC, we were able to return a valid solution that assigned values to unbounded variables. For the scenarios where no solution was found, we were able to identify the source of the inconsistency (POC). To our knowledge this is the first time inconstancies in concurrently used CPGs were automatically identified and flagged for further action by the physician.

The considered CPGs were quite simple. More complex CPGs may result not only in a larger number of variables, but also a larger number of constraints. Additionally the nesting and temporal dimension of a complex CPG can become issues that produce more variables. However, CLPs are solved for millions of variables and constraints in practice therefore these issues will not present a computational problem.

4 Discussion and Future Work

The approach presented in this paper is one of the few attempts to use CPG models as an active support tool that provides solutions for incomplete patient information and helps a physician identify inconsistencies in two (or more) CPGs for a patient with comorbidity. In our future work we will focus on representing more complex CPGs and developing external knowledge bases representing comprehensive interactions. We are exploring several ideas for improving the technique of revising CPGs according to identified POCs by using domain knowledge, thus significantly diminishing the need for revisions made by the physician.

Acknowledgements: Research described in this paper was funded by grants from the Natural Sciences and Engineering Research Council of Canada. The third author acknowledges support of the Polish Ministry of Science and Higher Education. The authors would like to thank Professor Brian Rowe, M.D. from the University of Alberta Hospital for help with interpreting clinical practice guidelines.

References

1. Dechter, R.: Constraint Processing. The MIT Press (1989)
2. Cabana, M.D., Rand, C.S., Powe, N.R., Wu, A.W., Wilson, M.H., Abboud, P.A.C., Rubin, H.R.: Why don't physicians follow clinical practice guidelines? a framework for improvement. Journ. of the American Medical Assoc. **282** (1999) 1458–1465
3. Latoszek-Berendsen, A., Tange, H., van den Herik, J., Hasman, A.: From clinical practice guidelines to computer-interpretable guidelines. Methods of Information in Medicine **49**(6) (2010) 550–570
4. Horvitz, E.: Automated reasoning for biology and medicine. In: Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, and Computer Vision. (1993)
5. Charles J. Petrie, J.: Revised dependency-directed backtracking for default reasoning'. In: AAAI-87. (1987) 167–172
6. Hing, M.M., Michalowski, M., Wilk, S., Michalowski, W., Farion, K.: Identifying inconsistencies in multiple clinical practice guidelines for a patient with co-morbidity. In: Proc. of KEDDH-10. (2010) 447–452
7. Lindblad, C., Hanlon, J., Ross, C., Sloane, R., Pieper, C., Hajjar, E., Ruby, C., Schmader, K.: Clinically important drug-disease interactions and their prevalence in older adults. Clinical Therapeutics **28**(8) (2006) 1133–1143