

Using Constraint Logic Programming for the Verification of Customized Decision Models for Clinical Guidelines

Szymon Wilk¹, Adi Fux², Martin Michalowski³, Mor Peleg², Pnina Soffer²

¹ Poznan University of Technology, Poznan, Poland

² University of Haifa, Haifa, Israel

³ MET Research Group, Ottawa, Canada

Abstract. Computer-interpretable implementations of clinical guidelines (CIG) add knowledge that is outside the scope of the original guideline. This knowledge can customize CIGs to patients' psycho-social context or address comorbidities that are common in the local population, potentially increasing standardization of care and patient compliance. We developed a two-layered contextual decision-model based on the PROforma CIG formalism that separates the primary knowledge of the original guideline from secondary arguments for or against specific recommendations. In this paper we show how constraint logic programming can be used to verify the layered model for two essential properties: (1) secondary arguments do not rule in recommendations that are ruled out in the original guideline, and (2) the CIG is complete in providing recommendation(s) for any combination of patient data items considered. We demonstrate our approach when applied to the asthma domain.

1 Introduction

Clinical practice guidelines (CPGs) include evidence-based recommendations intended to optimize patient care [1]. When physicians use CPGs in practice to make clinical decisions, they often take into consideration additional aspects related to the patient's personal context (e.g., the patient's level of family support, the degree to which his/her daily schedule are routine)[2]. Additional clinical aspects not contained in the CPG, such as comorbidities that the patient may have, are also considered during decision making. Typically there is no evidence-based recommendation for weighing in personal considerations into clinical decision-making. Moreover, CPGs cannot address all possible comorbidities that patients may have and such considerations are usually left to the discretion of the physician.

It follows that when a decision support system (DSS) is developed based on computer-interpretable clinical guidelines (CIGs) [9], it may be desirable to customize the CIG by also including arguments (conditions that provide support for and against specific recommendations) that are based on comorbidities and personal considerations which are common in the local settings. Customization aims to achieve more standard management by physicians, given that they better address relevant secondary considerations not mentioned in the CPG. Moreover, patients may be more compliant to recommendations that address their personal context [3].

Nevertheless, it is important to acknowledge that arguments associated with the

considerations not contained in the original CPGs should be secondary to the recommendations found in the CPGs, as they are generally not evidence-based. Hence any customized decision-making model should obey the *secondarity* property: *Secondary arguments should only modulate existing primary recommendations, while not suggesting recommendations that are not clinically indicated.* Modulation includes re-ranking of decision options, changes in dose or frequency of treatment or monitoring, changes in treatment or monitoring schedule, etc. Moreover, we would like the decision-model to obey the *completeness* property: *for any valid combination of primary and secondary decision-parameters (data items and results of previous decisions), at least one recommendation is indicated.* This property guarantees that the customized CIG will not encounter a situation where no valid candidate exists.

In this paper we introduce a two-layered contextual decision-model based on the PROforma CIG formalism [4] and operationalized within the Tallis enactment engine (<http://www.cossac.org/tallis>). We then present logic-based methods for verifying the secondarity and completeness properties of the two-layered model. We use a case-study from the asthma domain to demonstrate our approach.

2 Related Work

2.1 Customization of CPGs

Other researchers have developed methodologies that allow customizing CIG models so that they can be personalized at run time. Riaño et al.'s methodology [5] uses algorithms that manipulate domain ontologies to yield a personalized view of the healthcare knowledge to support clinical decisions for chronically ill comorbid patients. The methodology uses domain ontologies to provide decision support for adjusting a patient's condition based on disease profiles; these profiles are consulted to suggest additional signs and symptoms that the patient is likely to exhibit and which could be used to generate a more complete record. Grandi and coauthors [6, 7] suggest efficient management of multi-version CIGs collections by representing, in a knowledge base, multi-version clinical guidelines and domain ontologies in XML or in relational schemas. Personalized CIGs can be created by building from the knowledge base an on demand version that is tailored to the patient's current time (or desired temporal perspectives) and to the patient's disease profile (i.e., set of comorbidities). Finally, Michalowski et al. [8] expanded their mitigation framework based on first-order logic to account for patient preferences related to treatment. These preferences are represented in the form of preference-related revision operators that describe undesired circumstances (e.g., a sequence of treatment actions) that the patient would like to avoid, and specify changes that should be introduced to CPGs in order to make them consistent with patient preferences.

On one hand our work shares some similarities with the above approaches – the secondary layer can be seen as a very complex revision operator that expands the primary layer and brings additional data items into consideration. On the other hand, unlike other approaches, it explicitly verifies the validity of the obtained model to ensure it maintains the required properties.

2.2 Automatic Verification and Evaluation of CIGs

CIG verification techniques fall into three categories [9]: (1) proving that the CIG specification is internally consistent and free of anomalies, (2) proving that the CIG specification satisfies a set of desired formally defined properties, using model-checking or theorem-proving, and (3) checking inconsistencies between CIGs that are concurrently applied to a patient with comorbidities [10]. The approach presented in [11] uses two techniques: model checking to verify guidelines against semantic errors and inconsistencies in CIG definition, and model driven development to automatically process manually-created CIGs against temporal logic statements that should hold for these CIG specifications. Another technique is described in [12] where theorem proving explores logical derivations of a theory representing a CIG to confirm whether a formal CIG protocol complies with certain protocol properties.

Our approach relies on model checking – specifically we use constraint-logic programming (described in Section 3.2) to ensure that the required properties hold for a given two-layered decision model. Generally, model checking is easier and more efficient than theorem proving [13]. Moreover, applied techniques (e.g., constraint propagation) further facilitate representation and processing of CIGs.

3 Methods

3.1 Two-layered Contextual Decision Model

Following the definitions used in PROforma, a *plan (task network)* is a network composed of *tasks* and *scheduling constraints*. Tasks are specialized into a *plan*, *enquiry*, *action*, and *decision*. A decision has at least two *candidates* (or *recommendations*). Each *candidate* has at least one *argument* – a condition that refers to patient data items and an associated numerical weight (support) for or against the candidate. Our two-layered contextual decision-model extends PROforma’s CIG model by distinguishing between *primary* and *secondary* arguments. Primary arguments are formalizations of evidence-based recommendations found in a CPG and refer to *clinical (primary) data items*. Secondary arguments extend the CPG by constructing arguments that relate to additional *secondary data items* that are not part of the CPG. In our two-layered decision model, the *primary layer* is a plan where all arguments associated with decision candidates are primary arguments; the *secondary layer* includes a set of secondary arguments and their secondary data items that are associated with the decision candidates of a given primary layer. Weights for primary arguments correspond to the grades of evidence used by the CPG, while weights for secondary arguments are established by clinical experts based on their knowledge and experience.

Completeness is satisfied if for each decision there is at least one candidate with total support (i.e., sum of argument weights) in both layers that is equal to or greater than a threshold defined in the CIG; secondarity is satisfied if there is no decision candidate for which total support in the primary layer is lower than 0 and support is greater or equal to the threshold in both layers.

We created software that combines PROforma models representing primary and secondary layers into a single two-layered model. The software integrates the primary and

secondary arguments into their respective decisions. Upon enactment (using Tallis), all arguments are evaluated and decision candidates are ranked accordingly.

3.2 Constraint-Logic Programming and MiniZinc

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 [14]. A CLP model is made up of a set of variables with finite domains, a set of clauses with constraints, and a goal to be satisfied. The clauses in the model capture the relationships between variables and they restrict the possible combinations of values assigned to variables. 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 (i.e., bodies of all clauses are satisfied). It is also possible to expand the goal with a goal function and look for solutions that optimize it (maximize or minimize this function) while preserving all constraints. This is usually referred to as a constraint optimization problem (however, in this work we are not considering this variant).

There are specialized solvers for CSPs that employ various finite domain and linear programming techniques and use different and often incompatible modeling languages. MiniZinc is a medium-level constraint modeling language that has been widely accepted as a standard for CLP models [15] and we use it as our modeling language.

3.3 Using CLP to Check Properties of Two-layered Decision Models

In this study we use CLP to verify the completeness and secondarity of two-layered decision models and to control the process of introducing revisions necessary to ensure these properties. More specifically, decision models given in PROforma are translated into MiniZinc models, which in turn are verified and revised. Finally, the resulting MiniZinc models are translated back to PROforma.

Our overall goal is to ensure that a given PROforma model satisfies the properties of completeness and secondarity for all possible patient cases (i.e., all clinically valid combinations of primary and secondary data items). We achieve this goal indirectly by creating and solving a corresponding MiniZinc model with constraints to identify *problematic* cases that violate at least one of these properties. Thus, if there are no solutions to the MiniZinc model this indicates that there are no such cases and the validity of the underlying PROforma model has been positively verified.

Our approach is outlined in more details below:

1. Create a MiniZinc model from the initial PROforma model. The MiniZinc model contains three groups of variables:
 - Variables corresponding to primary and secondary data items defined in the enquiry steps (for each data item there is a unique variable),
 - Variables corresponding to *intermediate* decision candidates, i.e., these decision candidates that affect other decision candidates (for each intermediate candidate there is a unique variable),
 - Variables corresponding to support for specific decision candidates (for each candidate there are two unique variables corresponding to support in each layer).

Moreover, it contains the following groups of constraints:

- Constraints enforcing (or computing) support for individual candidates (for each variable corresponding to the support there is a unique constraint),
 - A single constraint that is a disjunction of two “sub-constraints” – one that enforces the violation of the completeness property, and the other that enforces the violation of the secondary property,
 - Optional constraints corresponding to domain knowledge that exclude combinations of variable values that represent clinically invalid solutions. Unlike the earlier constraints, the optional ones need to be specified manually by a clinical domain expert.
2. Solve the MiniZinc model. If there are no solutions, then go to step 4.
 3. Revise the MiniZinc model to avoid problematic cases by (1) modifying conditions and weights in existing arguments, (2) removing existing arguments and/or (3) adding new arguments, and then go to step 2. Revisions may be applied to both layers, thus following the principles of evidence-based medicine, we allow experts to adjust CPGs recommendations according to their experience.
 4. Translate the MiniZinc model to the final PROforma model focusing on constraints corresponding to arguments, as they are the ones that have been revised.

Currently revisions in step 3 are introduced manually using problematic patient cases from step 2 to direct the search for appropriate modifications, however, we plan to automate the revision step and thus to minimize the need for manual intervention.

3.4 Analysis of Property Violations and Revisions of the PROforma Model

We carried out a theoretical analysis of the reasons for violations of both properties and proposed corresponding revisions. The prevalent reason was incorrect argument weights. This calls for rescaling weights either in the primary or secondary layer to increase the difference between their orders of magnitude. Another reason was certain clinically infeasible combinations of data items that were initially missed by CIG modelers. These needed to be explicitly excluded. Introducing these revisions requires changing the clinical flowchart and updating of CIG both layers.

4 Case Study Example

We use an asthma guideline adapted from [16, 17] to which additional decision arguments based on personal context variables were added as a secondary layer. The additional arguments were based on interviews with 15 clinical experts from Israel. The asthma guideline starts with a decision regarding the clinical goal: an aggressive goal which tries to improve clinical indicators, or a basic goal, which tries to maintain their levels. Once a treatment goal is selected, three decisions are made regarding medication type (steroid or not), dose (high, medium, low), and treatment intervals (frequency: daily or weekly). Decision arguments in the clinical guideline refer to four clinical (primary) data items: number of monthly attacks (≤ 4 , 4-8, >8), severity of attacks (low, moderate, severe), forced expiratory volume (FEV1, which can be $<60\%$, 60-80%, $>80\%$) and daily limitation level (minor, medium, severe).

The secondary layer provides additional arguments for the existing decisions that are based on personal considerations. These include the routineness of the patient’s daily life (routine, semi-routine, no-routine), his/her communication level (low, medium, high) and his/her level of family support (frequent, medium, low).

5 Results

5.1 A Decision Model for Asthma with the Secondary Personal Domains

The two-layered model developed for this case study is given in Fig. 1a. The first decision in the model (“treatment_goal”) is an intermediate decision that does not lead directly to any action, but influences other decisions. Moreover, the model invokes an external plan for patients with a large number of monthly attacks (exceeding 20) as required by the asthma guideline. Fig. 1b displays arguments from the primary and secondary layers associated with the “aggressive_goal” decision candidate of the “treatment_goal” decision. For example, the aggressive goal is preferred for patients with high communications skills (who understand directions) and frequent family support (who may commit effort).

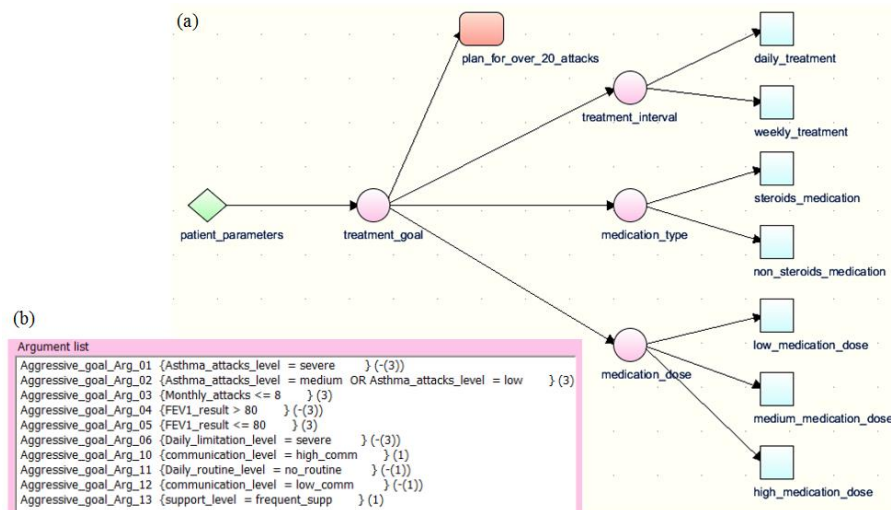


Fig. 1. (a) PROforma asthma CIG, where diamonds denote enquiry, circles – decisions, squares – actions and ovals – invocation of an external plan. (b) Primary (01..06) and secondary arguments (10..13) for the “aggressive_goal” candidate of the “treatment_goal” decision.

5.2 Verification and Revision of the Decision Model for Asthma

The PROforma model from Fig. 1 was the initial model for our verification and revision procedure described in the *Methods* section. We started by constructing a corresponding MiniZinc model – its representative parts are given in Figures 2 and 3.

Fig. 2 shows how we represent arguments (for brevity we focus on primary arguments, secondary ones are defined similarly) and how we compute support

associated with specific candidates in the primary and secondary layers. Specifically, each argument is a conditional *if...then...else...endif* expression associating conditions on data items with a weight. For brevity in the MiniZinc model we encoded symbolic values of specific data items with numbers (e.g., low, moderate, and severe attack levels are represented as 1, 2, and 3 respectively). A quick comparison of Figs 1b and 2 highlights the close correspondence between MiniZinc and PROforma.

```

var int: pl_aggressive_goal_support;
constraint
  pl_aggressive_goal_support =
    (if asthma_attacks_level == 3 then -3 else 0 endif) +
    (if asthma_attacks_level == 2 \\/ asthma_attacks_level == 1 then 3 else 0 endif) +
    (if monthly_attacks <= 8 then 3 else 0 endif) +
    (if FEV1_result > 80 then -3 else 0 endif) +
    (if FEV1_result <= 80 then 3 else 0 endif) +
    (if daily_limitation_level == 3 then -3 else 0 endif);

```

Fig. 2. Constraints that introduce arguments and calculate support for the “aggressive_goal” candidate in the primary layer.

For brevity, Fig. 3 presents selected parts of the constraint that enforces the violation of the completeness and secondarity properties. It is formulated over variables representing support for specific decision candidates. All support thresholds in the initial PROforma model are equal to 1.0, and the same value was used in the MiniZinc model. The constraint in Fig. 3 enables the solver to identify problematic patient cases for whom these properties are violated. Moreover, the MiniZinc model contains constraints to eliminate solutions that are clinically invalid (e.g., that combine more than 8 monthly attacks with minor daily limitation level). The structure of these latter constraints is relatively simple and as such we do not present them here.

```

constraint
  ((pl_plan_for_over_20_attacks_support + sl_over_20_attacks_support_plan_support < 1 /\
   pl_aggressive_goal_support + sl_aggressive_goal_support < 1 /\
   pl_basic_goal_support + sl_basic_goal_support < 1) \\/ [...]) (a)
  \\/
  ((pl_plan_for_over_20_attacks_support < 0 /\
   pl_plan_for_over_20_attacks_support + sl_plan_for_over_20_attacks_support >= 1) \\/ [...]) (b)

```

Fig. 3. Disjunctive constraint that identifies patient cases that violate the completeness property (a) or the secondarity property (b) in the two-layer model. Prefixes “pl_” and “sl_” identify variables associated with primary and secondary layers, respectively

The initial MiniZinc model was solved and the solver found 28,656 solutions violating any of the two considered properties. This large number was caused by numeric primary data items leading to thousands of possible combinations of their values (a pre-discretization of numerical data items could have addressed this problem). Interestingly, all these solutions violated only the completeness property; there was no solution that violated secondarity. In Table 1 we present examples of problematic patient cases found at this stage. We performed a more detailed analysis of these patient cases to identify specific decisions for which completeness was violated – they are listed in the last row of this table.

Because of the identified problematic patient cases, the MiniZinc model had to be revised. Revisions were introduced by a knowledge engineer, who worked with a domain expert (physician). The knowledge engineer focused on primary decision arguments. For each specific violation (associated with a specific decision or candidate)

he identified the reason and then proposed several fixes (i.e., modifications of the argument list) that were vetted by the expert (the expert was also able to provide his own corrections). The most appropriate fix was introduced to the MiniZinc model. After the first round of revisions, the solver still found solutions indicating problematic patients, thus the revision and verification steps needed to be repeated. Overall, it took 9 iterations to arrive at the final MiniZinc model where no patient cases violating any of the properties were found. The knowledge engineer and the domain expert accepted this model and it was translated to the final PROforma model.

Data item	Case 1	Case 2	Case 3
asthma_attack_level	low	moderate	low
daily_limitation_level	severe	minor	minor
FEV1_result	81	81	81
monthly_attacks	5	1	5
daily_routine_level	no_routine	semi_routine	routine
communication_level	medium_comm	medium_comm	medium_comm
support_level	low_supp	medium_supp	medium_supp
Decision with violated completeness	treatment_goal, treatment_interval, medication_type	treatment_interval	medication_type

Table 1. Sample problematic patient cases violating the completeness property of the two-layered PROforma model.

The differences between the initial and final PROforma models are summarized in Table 2. The changes were focused on the three decisions in the primary layer – “treatment_goal”, “treatment_interval” and “medication_type.” The most extensive revisions were associated with candidates of the first decision, where more than 20 new arguments were added to the model. Significant changes were also introduced for the “steroids” candidate of “medication_type”. On the contrary, no revisions were made to the secondary layer and to the “medication_dose” decision in the primary layer.

Decision	Candidate	# Arguments					
		Primary Layer			Secondary Layer		
		Initial model	Final model	Change	Initial model	Final model	Change
treatment_goal	basic_goal	4	27	23	2	2	0
	aggressive_goal	6	37	31	4	4	0
	plan_for_over_20_attacks	1	1	0	0	0	0
treatment_interval	weekly_interval	7	18	11	3	3	0
	daily_interval	4	20	16	1	1	0
medication_type	non_steroids	4	14	10	1	1	0
	steroids	6	25	19	2	2	0
medication_dose	low_dose	6	6	0	3	3	0
	medium_dose	5	5	0	2	2	0
	high_dose	8	8	0	2	2	0

Table 2. Differences between the initial and final PROforma models in terms of the number of arguments associated with specific decision candidates.

In Fig. 4 we present arguments associated with the “non_steroids” candidate of the “medication_type” decision in the initial and final PROforma models. The introduced revisions not only modified weights or expanded conditions in existing arguments

(compare argument 03 in Fig. 4a and 10 in Fig. 4b, or argument 04 in Fig. 4a and arguments 02...04 and 11..12 in Fig. 4b), but also removed existing arguments (01 in Fig. 4a) and added multiple new ones. We note that the list of arguments in the final PROforma model could be shortened by combining some arguments (e.g. 08 and 09 in Fig. 4b). However the extent of changes would still be significant.

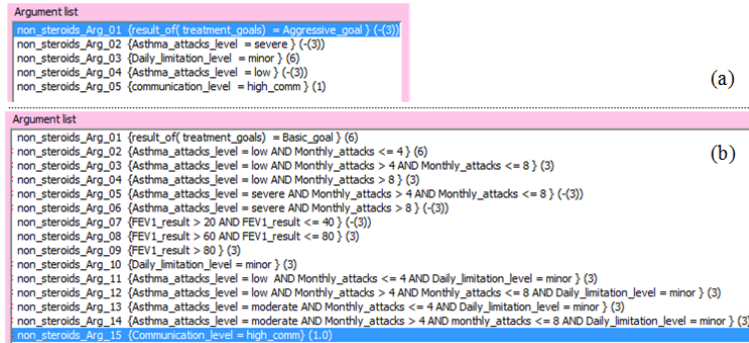


Fig. 4. Arguments associated with the “non_steroids” candidate of the “medication_type” decision in the initial PROforma model (a) and in the final PROforma model (b).

6 Discussion and Conclusions

In this paper we demonstrated that the applicability of CLP to two-layered decision models allows for their automatic verification and suggests areas of focus for revision, saving a knowledge engineer significant manual work. Layered models can become quite complex and although the knowledge engineer who created the models is experienced, he/she could not manually find all errors in the model within a reasonable amount of time. All errors were exposed using the CLP verification approach by iteratively checking the MiniZinc models for satisfiability allowing for easier construction of more complex two-layered models. We plan to further expand this approach to automatically find the appropriate weights for primary and secondary arguments.

By encoding the guideline as a set of constraints in the model, the argument weights as variables, and negating the current version of the secondarity and completeness verification constraint, our approach can solve this extended model for feasibility. A feasible solution represents an assignment of weights for all arguments such that secondarity and completeness are guaranteed in the decision model. In this work we focused on one way of modulating the recommendations provided by the primary layer, specifically modulating the support for different candidates by changing their ranking. Our secondary layer also only considered the psycho-social context of patients.

Future research will examine the use of CLP for verifying PROforma models where modulation also involves changes in dose or frequency of treatment or monitoring, changes in treatment or monitoring schedule, etc., and where the secondary context includes other factors such as additional comorbidities and local setting (e.g., organizational resources, local regulations).

7 References

1. Institute of Medicine, Clinical Practice Guidelines We Can Trust (2013).
2. Fux, A., Peleg, M., Soffer, P.: How Does Personal Information Affect Clinical Decision Making? Eliciting Categories of Personal Context and Effects. *AMIA Annu Symp*, 1741 (2012).
3. Quinn, C.C., Gruber-Baldini, A.L., Shardell, M., Weed, K., Clough, S.S., Peeples, M., Terrin, M., Bronich-Hall, L., Barr, E., Lender, D.: Mobile diabetes intervention study: testing a personalized treatment/behavioral communication intervention for blood glucose control. *Contemp Clin Trials* 30 (4), 334-46 (2009).
4. Sutton, D.R., Fox, J.: The syntax and semantics of the PROforma guideline modeling language. *J Am Med Inform Assoc* 10 (5), 433-43 (2003).
5. Riano, 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 Biomed Inform* 45 (3), 429-46 (2012).
6. Grandi, F.: Dynamic class hierarchy management for multi-version ontology-based personalization. *Journal of Computer and System Sciences* 82 (1, Part A), 69-90 (2016).
7. Grandi, F., Mandreoli, F., Martoglia, R.: Efficient management of multi-version clinical guidelines. *J Biomed Inform* 45 (6), 1120-36 (2012).
8. Michalowski, M., Wilk, S., Rosu, D., Kezadri, M., Michalowski, W., Carrier, M.: Expanding a First-Order Logic Mitigation Framework to Handle Multimorbid Patient Preferences. *AMIA Annu Symp Proc* 2015, 895-904 (2015).
9. Peleg, M.: Computer-interpretable clinical guidelines: a methodological review. *J Biomed Inform* 46 (4), 744-63 (2013).
10. Wilk, S., Michalowski, W., Michalowski, M., Farion, K., Hing, M.M., Mohapatra, S.: Mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming. *J Biomed Inform* 46 (2), 341-53 (2013).
11. Perez, B., Porres, I.: Authoring and verification of clinical guidelines: a model driven approach. *J Biomed Inform* 43 (4), 520-36 (2010).
12. ten Teije, A., Marcos, M., Balsler, 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).
13. Halpern, J.Y., Vardi, M.Y.: Model checking vs. theorem proving: a manifesto. In: Vladimir, L., (ed.) *Artificial intelligence and mathematical theory of computation*, pp. 151-176. Academic Press Professional, Inc. (1991).
14. Dechter, R., *Constraint Processing*. MIT Press (1989).
15. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: Towards a standard CP modelling language. In: Bessiere, C., (ed.) *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming*, pp. 529-543. Springer (2007).
16. British Thoracic Society and Scottish Intercollegiate Guidelines Network, QRG 141 - British Guideline on the Management of Asthma (2014).
17. Israel Medical Association: Clinical Practice Guidelines for Asthma Management [in Hebrew]. Harefuah (2000).