Using Model Checking for Critiquing based on Clinical Guidelines

Perry Groot [a,*] Arjen Hommersom [a] Peter J.F. Lucas [a] Robbert-Jan Merk [b] Annette ten Teije [b] Frank van Harmelen [b] Radu Serban [b]

[a] *Institute for Computing and Information Sciences, Radboud University Nijmegen, the Netherlands*
[b] *Division of Mathematics and Computer Science, Vrije Universiteit Amsterdam, the Netherlands*

Abstract

Objective: Medical critiquing systems compare clinical actions performed by a physician with a predefined set of actions. In order to provide useful feedback, an important task is to find differences between the actual actions and a set of 'ideal' actions as described by a clinical guideline. In case differences exist, the critiquing system provides insight into the extent to which they are compatible.

Methods and Material: We propose a computational method for such critiquing, where the ideal actions are given by a formal model of a clinical guideline, and where the actual actions are derived from real world patient data. We employ model checking to investigate whether a part of the actual treatment is consistent with the guideline.

Results: We show how critiquing can be cast in terms of temporal logic, and what can be achieved by using model checking. Furthermore, a method is introduced for off-line computing relevant information which can be exploited during critiquing. The method has been applied to a clinical guideline of breast cancer in conjunction with breast cancer patient data.

Key words: Critiquing, Clinical guidelines, Model-checking, Formal methods

1. Introduction

At the moment clinical guidelines are mostly just documents consisting of text, tables, and graphics. There is an increasing interest amongst researchers to develop computerised versions of such guidelines, using one of the specialised guideline representation languages. The resulting computer-based guidelines can then act as a basis for the development of decision-support systems, which then allow computer-based deployment of guidelines in a clinical setting. One possible application of such clinical decision-support systems is *critiquing*, i.e., to spot and analyse differences between the proposed actions taken by a medical doctor, and a set of 'ideal' actions as prescribed by the computerised guideline. As a computer-based clinical guideline is represented in a formal language, there is also room for a formal underpinning of the various ways a guideline can be manipulated.

A natural way to formally describe the actions taken by a medical doctor in the management of the disease of a patient is offered by temporal logics. As a family of languages, logics make it possible to describe the meaning of the various aspects of the disease and condition of the patient in a precise fashion. By the addition of temporal operators, *temporal* logic adds various notions of progress of the disease and sequencing of actions in time.

One way to look upon a patient and a patient's disease formally is as a concurrent system, i.e., as a system described in terms of states and state transitions in time. Model checking technology offers methods that allow one to analyse concurrent systems for their consistency. One can rely on an extensive collection of tools and techniques readily available. Model checking is a well investigated technique for verification of systems that can be modelled by a finite transition system. Model checking, however, has been mainly applied to technical systems, such as hardware, software-based communication protocols, concurrent programs, etc. This raises the central question of this paper: *can model checking be used as a basis for critiquing in medicine*?

Model checking takes as input a state transition system and a specification. In this case, a formalised guideline is taken as the state transition system, the actions that have

[*] Corresponding author. Tel.:+31 24 3652075, Fax:+31 24 3653366.
Email addresses: perry@cs.ru.nl (Perry Groot), arjenh@cs.ru.nl (Arjen Hommersom), peterl@cs.ru.nl (Peter J.F. Lucas), rjmerk@xs4all.nl (Robbert-Jan Merk), annette@cs.vu.nl (Annette ten Teije), frank.van.harmelen@cs.vu.nl (Frank van Harmelen).

been performed on a specific patient form the logical specification. Model checking then involves investigating the consistency between the formalised guideline and the treatment administered to the patient.

The innovative part of this work is the exploration of the use of model checking in the analysis of medical knowledge (guidelines and patient data) for the purpose of critiquing.

The structure of this paper is as follows: in Section 2, we outline our general approach and provide a categorisation of different types of non-compliance; Section 3 briefly outlines the essential logical foundations. Sections 4 and 5 are the core sections of this paper, where we describe two ways in which model-checking can be deployed for critiquing. The final Sections 6 and 7 compare the pros and cons of the methods, which are related to other related work in this area.

2. Approach and types of non-compliance

The central feature of a critiquing system is that the user of the system provides as input (1) a problem description (e.g., patient symptoms), and (2) a particular solution (e.g., treatment proposed). This second input distinguishes critiquing systems from the more traditional expert systems, which only take a problem description as input [22, 8]. The second input to a critiquing system, i.e., a particular treatment, is typically the expert system's output.

In our approach to critiquing medical treatments using model checking, the input to the system consists of a patient description and a treatment description. The patient description consists of patient symptoms and test outcomes measured for the patient, whereas the treatment consists of all actions (to be) performed by the practitioner. Both patient description and treatment description are typically provided by electronic records.

In our approach these are given to the system as temporal logic formulas. The appropriate form for such formulas is extensively discussed in Sections 4 and 5.

The critiquing system uses the patient and treatment descriptions as specifications that need to be checked against a formal model of the guideline. Such a formal model of a guideline can be viewed as a state transition system. Several notions of compliance of a treatment can then be considered. One definition of compliance is obtained by comparing actual sequences of actions, i.e., the treatment, with those mentioned in the clinical guideline. In addition, one could check whether all actions, when performed, yielded the expected outcome. In terms of guideline-modelling languages such actions are called 'successful'. Not always are the results of such actions explicitly mentioned in electronic patient records, in particular this is not the case if such records have been obtained as part of an epidemiological registry, such as a national cancer registry.

Types of non-compliance

We give a categorisation of different types of non-compliance. A number of reasons for non-compliance between the patient data, which includes clinical and laboratory findings and the actual treatment, on the one hand, and the treatment and findings required to select the treatment according to the guideline, on the other hand, are possible. As suggested above, two types of non-compliance are being distinguished. The first, 'T' type focuses on identifying differences between *actual* treatments and treatments suggested by the guideline, both consisting of sequences of actions. In contrast, the second, 'F' type considers the required *patient findings*, clinical or laboratory-based, for choosing a particular action that is part of a treatment. All cases of non-compliance offer reasons for scrutinising the patient data, and therefore the clinician's management of the patient's disorder, assuming that the clinical guideline is correct. Of course, in clinical practice there can be good reasons to deviate from the guideline, so the output of a critiquing system can only alert for *potential* problems.

Possible reasons for a *non-compliant treatment* are the following:

T1–T2 Non-compliant actions:
According to the guideline and the patient findings, some of the actions of the actual treatment were chosen wrongly. The following reasons are being distinguished:

- **T1: Action unsupported by patient findings:** There is at least one finding that was required for deciding about the action. For instance, the guideline suggests that the treatment 'adjuvant hormonal therapy' should only be given when the tumour contains estrogen hormone receptors. Suppose that the treatment of the patient includes the action 'prescribe adjuvant hormonal therapy' although hormone receptors have not been determined. Then, the action should not have been undertaken.
- **T2: Conflicting actions:** There is a mismatch between the patient findings and the required findings, according to the guideline, for the treatment that has been given. For example, the guideline says that the hormone receptor status should be positive in order to apply adjuvant hormonal therapy. Consequently, patients that are hormone-receptor negative should not have received this treatment. There is a conflict between the actual action and the appropriate action according to the guideline.

T3: Non-compliant order of actions:
The actions constituting the treatment are right; however, the order in which they are performed is wrong. For example, according to the guideline, for a certain patient group 'surgery' should be followed by 'radiotherapy'; the actual treatment was 'radiotherapy' followed by 'chemotherapy'.

T4: Missing mandatory actions:
Another possibility is that, according to the guideline and

the patient findings, a particular action should be part of the treatment. However, it is missing from the patient data. An example is that all required conditions for 'adjuvant hormonal therapy', such as a 'positive hormone receptor status', are included in the patient findings, where 'adjuvant hormonal therapy' was not included in the actual treatment.

Whereas T compliance compares the actions included in patient data with those in the guideline, and critiques the choice of these actions, F compliance of patient findings takes the opposite viewpoint that the actual treatment is correct and that differences between patient data and the guideline can be explained in terms of patient findings. The following non-compliances are being distinguished:

F1: Missing relevant findings for an action:
Here the right action was taken according to the guideline; however, the patient data lacks findings that support that action. For example, the guideline mentions that the choice of the action radiotherapy depends on tumour size, whereas the patient has been treated by the radiotherapy without having information about tumour size available.

F2: Wrong findings for an action:
There exists a mismatch between the patient findings and the findings required for the choice of an action according to the guideline. For example, 'adjuvant hormonal therapy' is part of the treatment where the 'hormone receptor status' is negative, whereas, according to the guideline the 'hormone receptor status' should have been positive.

F3: Non-compliant findings for the order of actions:
The findings associated with the order of actions in the actual treatment are not compatible with those required according to the guideline. For example, chemotherapy is given before surgery in order to reduce the tumour load to ensure that surgery is more likely to yield a successful outcome. In other cases, chemotherapy is applied after surgery in order to reduce the risk of recurrence. The patient findings are incompatible with the first possibility, which nevertheless was chosen.

F4: Containing irrelevant findings:
The patient findings include redundancy, i.e., findings which are not referred to in the guideline. For instance, a laboratory blood test was done but its results is ignored in the selection of treatment.

Note that T1-F1, T2-F2, T3-F3 are related to each other: either the chosen action is wrong in the context of the patient findings, or the patient findings are incompatible with the selected treatment.

In Section 4 and Section 5 we describe two methods for identifying those types of non-compliance. Both methods are using model checking techniques for critiquing a patient record against a guideline. The method described in Section 4 uses model checking in a *direct* way: the guideline and the patient record are the input of the model checker and the output is whether the patient record is compliant with the guideline. The method described in Section 5 uses model checking in an *indirect* way: the model checker is used for constructing constraints, and then these constraints are checked against the patient record. *Both* critiquing methods are able to handle both types of non-compliant treatment and non-compliant patient findings. Both approaches lead to further insight into the nature of the detected inconsistency allowing the system to exploit these insights into a critique, which is then given to the practitioner.

3. Preliminaries

Subsection 3.1 introduces the formal preliminaries of temporal logic. Subsection 3.2 discusses the state transition system representation used for modelling the guideline.

3.1. *Temporal logic*

Temporal logic deals with the representation of temporal information within a logical framework. Here, we take a modal-logic approach, where relationships between worlds in the usual possible-world semantics of modal logic is understood as time order. Properties of guidelines are specified using both Computation Tree Logic (CTL) and Linear Temporal Logic (LTL) [7], either on their own or using modular model checking [11] (e.g., Equation (9)).

In this paper we model a guideline as a Kripke structure M over a set of atomic propositions AP, which formally is defined as a four tuple $M = (S, s_0, R, L)$ where S is a finite set of states, $s_0 \in S$ is the initial state, $R \subseteq S \times S$ is a total transition relation, and $L : S \to 2^{AP}$ is a function that labels each state with the set of atomic propositions true in that state. A *path* in the model M from a state s is an infinite sequence $\pi = s_0 s_1 s_2 \ldots$ such that $s_0 = s$ and $R(s_i, s_{i+1})$ holds for all $i \geq 0$. With π^i we denote the suffix of π starting at s_i, i.e., $\pi^i = s_i s_{i+1} s_{i+2} \ldots$.

CTL uses atomic propositions, propositional connectives, *path quantifiers*, and *temporal operators* for describing properties of *computation trees*, i.e., the tree that is formed by designating a state in the Kripke structure as the initial state and then unwinding the structure into an infinite tree according to the transition relation R with the initial state as root. This leads to two types of formulas: *state formulas*, which are true in a specific state, and *path formulas*, which are true along a specific path. A path formula is build up by applying one of the temporal operators to one or two state formulas. In this paper, the temporal operators used are $\mathbf{X}$, $\mathbf{G}$, $\mathbf{F}$, and $\mathbf{U}$. With $\mathbf{X}\varphi$ being true if φ holds in the next state, $\mathbf{G}\varphi$ if φ holds in the current state and all future states, $\mathbf{F}\varphi$ if φ holds in the current state or some state in the future, and $\varphi\mathbf{U}\psi$ if φ holds until eventually ψ holds. A state formula can be built inductively from atomic propositions, propositional connectives, and if f and g are path formulas, then $\mathbf{E}f$ and $\mathbf{A}f$ are state formulas. The path quantifiers $\mathbf{A}$ and $\mathbf{E}$ are used to specify that all or some of the paths starting at a specific state have some property.

$$
\begin{array}{ll}
M, s \models p & \Leftrightarrow p \in L(s) \\
M, s \models \neg f_1 & \Leftrightarrow M, s \not\models f_1 \\
M, s \models f_1 \wedge f_2 & \Leftrightarrow M, s \models f_1 \text{ and } M, s \models f_2 \\
M, s \models \mathbf{E} g_1 & \Leftrightarrow \text{there exists a path } \pi \text{ from } s \text{ such that} \\
& \quad M, \pi \models g_1 \\
M, \pi \models f_1 & \Leftrightarrow s \text{ is the first state of } \pi \text{ and } M, s \models f_1 \\
M, \pi \models \neg g_1 & \Leftrightarrow M, \pi \not\models g_1 \\
M, \pi \models g_1 \wedge g_2 & \Leftrightarrow M, \pi \models g_1 \text{ and } M, \pi \models g_2 \\
M, \pi \models \mathbf{X} g_1 & \Leftrightarrow M, \pi^1 \models g_1 \\
M, \pi \models \mathbf{F} g_1 & \Leftrightarrow \text{there exists } k \geq 0 \text{ such that } M, \pi^k \models g_1 \\
M, \pi \models g_1 \mathbf{U} g_2 & \Leftrightarrow \text{there exists } k \geq 0 \text{ such that } M, \pi^k \models g_2 \\
& \quad \text{and for all } 0 \leq j < k, M, \pi^j \models g_1
\end{array}
$$

Fig. 1. Semantics of temporal logic with f_1 and f_2 representing state formulas and g_1 and g_2 representing path formulas.

The semantics of CTL is defined with respect to a Kripke structure M. Given a state formula f, the notation $M, s \models f$ denotes that f holds in state s of the Kripke structure M. Assuming that f_1 and f_2 are state formulas and g_1 and g_2 are path formulas, the relation $\models$ is defined inductively as shown in Fig. 1. The remaining syntax consisting of $\vee, \rightarrow, \mathbf{G}, \mathbf{A}$ can be defined as usual, i.e., $f_1 \vee f_2 \equiv \neg(\neg f_1 \wedge \neg f_2)$, $f_1 \rightarrow f_2 \equiv \neg f_1 \vee f_2$, $\mathbf{G} g \equiv \neg \mathbf{F} \neg g$, and $\mathbf{A} f \equiv \neg \mathbf{E} \neg f$.

In contrast to CTL, LTL provides operators for describing events along a single computation path. Each formula is of the form $\mathbf{A} f$, with f being a path formula, which is either an atomic proposition or inductively defined as $\neg f$, $f \vee g$, $f \wedge g$, $\mathbf{X} f$, $\mathbf{F} f$, $\mathbf{G} f$, $f \mathbf{U} g$ with f, g path formulas. This language can be evaluated on Kripke structures presented in Fig. 1.

3.2. *State transition system*

The guideline is modelled in the Asbru guideline modelling language [21]. A medical guideline is considered in Asbru as a hierarchical *plan*, where each plan can consist of actions, sub-plans, or both to be executed as part of the plan. The semantics of Asbru is defined in [4] by flattening the hierarchy of plans and using one top level control to execute all plans synchronously. Within each top level step, a step of every plan is executed. Whether a plan is able to progress depends on its conditions. Plans progress from a selection phase, to an execution phase, and finally a termination phase. A plan can either terminate successfully (complete) or unsuccessfully (abort), after which it remains in this state, unless the plan is re-selected for execution.

The Asbru model is translated to a state transition system using the techniques and tools as described in [5]. Each path in this state transition system can be considered a patient who is given a treatment that is consistent with the recommendation described by the guideline. Patient descriptions, e.g., the findings, are modelled in each state as a parameter-value combination. In the first state of the model all the patient parameters are *unknown*, whereas in the second state the parameters are initialised non-deterministically. For simplicity, we will assume that the parameters do not dynamically change during the execution of the plan, i.e., the parameters describe the patient before any interventions have been performed.

If an action is applied to a certain patient group, then it should be possible, according to the guideline, to complete this action successfully. Therefore, when we say that an action α holds in a certain state of the model, we mean that it can be completed, which is formalised as:

$$\text{state}(\alpha) \neq \text{completed} \wedge \mathbf{EX}\, \text{state}(\alpha) = \text{completed} \tag{1}$$

While it is possible to argue for other choices (e.g., activation, or both activation and completion), the advantage of this definition is that actions that necessarily abort are not taken into account.

4. Direct critiquing

In this section, we investigate, by directly giving the patient record and treatment plan to the model checker, how we can check properties of the guideline model in order to generate critiques about the proposed treatment.

4.1. *Critiquing formulas*

4.1.1. *CTL formulas*

In order to formalise the different types of non-compliances, two concepts are central. The first describes whether an action α is in *conflict* with the patient findings P, i.e., whether:

$$\mathbf{AX}\,(P \rightarrow \mathbf{AG}\,\neg\alpha) \tag{2}$$

holds, where $\mathbf{AX}$ is used to refer to the state where the patient data is initialised. This property denotes that for the findings described by P, α should definitely not be applied. The second concept describes a more strict relation between the action and patient, namely whether the action is *supported* by the patient findings, i.e., whether the action should always be done for the patient, i.e.,

$$\mathbf{AX}\,(P \rightarrow \mathbf{AF}\,\alpha) \tag{3}$$

If an action is supported for a patient that is contained in one of the treatment paths of the model, then it cannot be in conflict. However, it is possible that the action is neither in conflict nor supported, i.e., if the action is optional.

From these two concepts, we can build up the non-compliances. Non-compliance F1 can be investigated by establishing that for the patient findings P, α is not supported and for some extension of P, α is supported. This yields the following formula:

$$\neg\mathbf{AX}\,(P \rightarrow \mathbf{AF}\alpha) \wedge \mathbf{AX}\,(P \cup P' \rightarrow \mathbf{AF}\alpha) \tag{4}$$

where P' is a consistent set of parameter-value combinations disjoint of P. Each minimal P' can be considered a set of missing relevant findings. For example, if it is necessary to know the hormone level to support hormonal therapy, then P' would contain this parameter with the necessary value. Even though this also establishes non-compliance T1, if we are only interested in establishing that the action is not supported then we can suffice with:

$$\neg\mathbf{AX}\,(P \rightarrow \mathbf{AF}\alpha) \wedge \mathbf{EX}(P \wedge \mathbf{AF}\alpha) \tag{5}$$

i.e., there is a set of findings consistent with P so that α is supported. This, however, does not provide the user with the actual set of parameters that supports α.

Non-compliances T2 and T3 describe whether or not an action, or sequence of actions are in conflict. The formula to check non-compliance T2 was shown in Equation (2), or equivalently, whether its negation is false, i.e., that

$$\mathbf{EX}(P \wedge \mathbf{EF}\alpha) \tag{6}$$

is false. For a sequence of actions $\alpha_1, \alpha_2, \ldots$, this can be generalised to:

$$\mathbf{EX}(P \wedge \mathbf{EF}(\alpha_1 \wedge \mathbf{EX}\,\mathbf{EF}\,(\alpha_2 \wedge \mathbf{EX}\,\mathbf{EF}\,(\ldots))) \tag{7}$$

i.e., in some treatment α_1 is done, and after that α_2, etc. If this formula is false, then we have either non-compliance T2 or T3. Recursive selection of subsets and relaxation of the ordering constraints can shed more light onto this. This will be further elaborated in Section 4.2.

For non-compliances F2 and F3, consider subsets P' of P. In case an action α is not in conflict for P' while $P' \cup \{p\}$ for some $p \in P$ is in conflict, then $P \setminus P'$ are incorrect findings for P. Non-compliance F3 is similar but it uses sequences rather than individual actions.

For checking that we are missing a mandatory plan, i.e., non-compliance T4, we can check for all actions α that are not part of the treatments proposed by the physician, whether it holds if α is supported. If it is, then we know that α is a missing mandatory action.

Finally, non-compliance F4 can be found by checking which findings can be removed taken into account that the action should remain mandatory, i.e., by checking whether $P \setminus \{p = v\}$, where $(p = v) \in P$, supports the action α. If this is true, then apparently $p = v$ is irrelevant for supporting the action α.

4.1.2. *LTL formulas*

In general, CTL model checking is more efficient than LTL model checking. However, for checking the compliance of a certain treatment of which we do not know the order between the individual actions, a CTL formula consists of a disjunction of each possible order of actions and considers the existence of each order. In case of n actions, with all order unknown, this leads to formulas of size $O(n \times 2^n)$. Similarly, when global properties of the treatment path are introduced, for example the state of the patient or the fact that some action *never* occurs, such knowledge becomes difficult to express. Assume for example a global property described by β, then Formula (7) must be rephrased to the rather complicated formula:

$$\mathbf{E}(\beta\,\mathbf{U}\,(\alpha_1 \wedge \beta \wedge \mathbf{EX}\,\mathbf{E}(\beta\,\mathbf{U}\,(\alpha_2 \wedge \beta \wedge \\ \mathbf{EX}\,\mathbf{E}(\ldots \wedge\ \mathbf{EG}\,\beta))))) \tag{8}$$

i.e., β holds until at some point α_1 and β (still) holds, after which β holds, etc.

Usually, the knowledge is reasonably complete and the global information is sparse, however, for a more succinct representation we can either use a more expressive logic such as CTL* [7] or consider LTL model checking. An approach using LTL is modular model checking [11], where the model is restricted using an LTL formula to those traces where the formula is valid. To prove the *existence* of a treatment in this approach, it is required to verify that the model restricted to a specification of a certain patient and treatment is not empty. Let φ be an LTL formula and $[\varphi]M\langle\bot\rangle$ denote that the set of LTL assertions φ leads to an empty model, i.e., φ describes a trace not present in the model. In contrast, if $[\varphi]M\langle\bot\rangle$ is shown to be false, then M can not be empty when restricted to φ proving that the trace described by φ exists in the model M. Formula (8) can thus be verified by showing that

$$[\mathbf{A}(\mathbf{G}\beta \wedge \mathbf{F}(\alpha_1 \wedge \mathbf{XF}(\alpha_2 \wedge \ldots)))]M\langle\bot\rangle \tag{9}$$

is *false*. An additional benefit of this representation is that when order information is absent, the property is typically more intuitively specified. Nonetheless, when there are few actions involved and much of the order information is present, CTL formulas are expected to be more efficient to verify.

4.2. *Critiquing method*

The critiquing method is described in Fig. 2. First, we check the physician's compliance by verifying consistency of the treatment performed with the patient data using the guideline model. This is repeated iteratively by relaxing the constraints on the model. The critique can then be further refined by considering subsets of the findings (to identify irrelevant findings) and actions (to find unsupported actions) and supersets of the findings (to identify relevant findings) and actions (to find mandatory actions). Typically, there are different choices to be made how the critique is built up, e.g., one could report that an action is not or that some finding is missing. Here, we have decided to abstract from these choices and look at the critiquing from a more high-level point of view.

In the remainder of this section, a part of this procedure, namely checking which part of the input is treatment

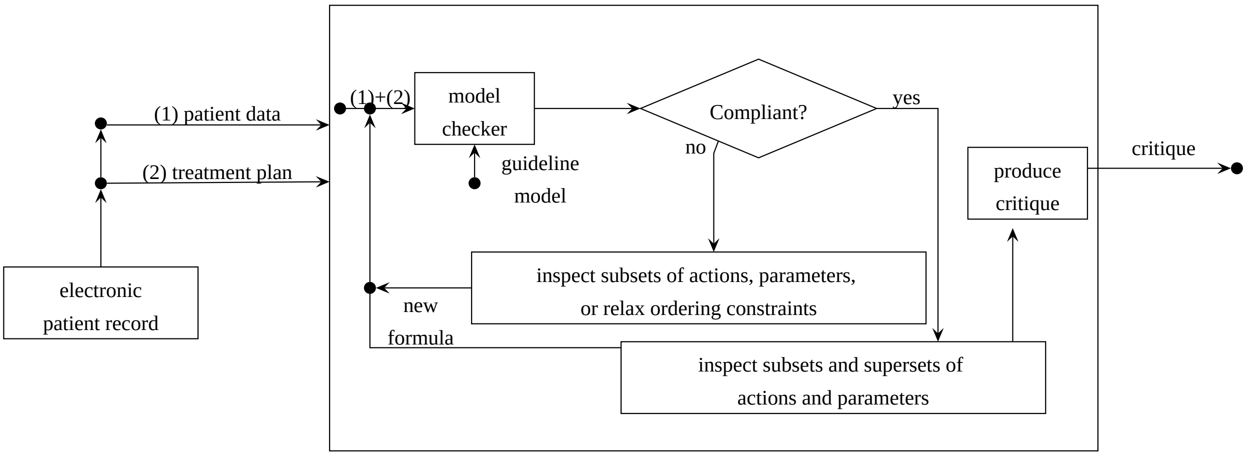
(1) patient data
(2) treatment plan
electronic patient record
(1)+(2)
model checker
guideline model
Compliant?
yes
no
produce critique
critique
inspect subsets of actions, parameters, or relax ordering constraints
new formula
inspect subsets and supersets of actions and parameters

Fig. 2. Critiquing approach using model checking. Given patient data and a treatment plan as input (temporal specifications), the critiquing system uses a model checker to verify consistency with respect to a guideline model (state transition system) to generate a critique (empty in case of compliance).

compliant (i.e., compliant according to T2 and T3), is illustrated using a case study.

4.3. *Application of the methodology to breast cancer*

4.3.1. *Design and choice of case studies*

The clinical guideline used is the Dutch breast cancer guideline[1] and was represented as a state transition system in Cadence SMV using the techniques and representation described in [5]. The models used here were developed as part of the Protocure-II project.[2] Patient data were obtained from the Dutch Comprehensive Cancer Centre South (CCC), a registry in the Netherlands used for cancer research, planning of services, and evaluation and implementation of guidelines. The data collected concerns breast cancer patients treated in the period January 2003 - June 2004, when the guideline was applicable, and therefore suitable for compliance checks with the guideline. Each patient record consists of 269 variables, which includes information about the diagnosis and treatment.

The patient data from the registry could, in principle, directly be used for critiquing with respect to the guideline. However, matching such data records to the terminology of the guideline is hard [14] and differs from the course commonly followed in medicine. In medical literature, specific patient cases, called *casuistics*, are frequently discussed in detail to gain insight into the way the patient's disease was managed. These papers follow a long standing tradition and are seen as part of the 'education permanente' of the medical profession. Critiquing in this paper was therefore done casuistically by having the CCC patient data interpreted by medical experts who provided a direct mapping from the patient data in the registry to the guideline. Subsection 4.3.2 presents in more detail a case-study in critiquing using the casuistic interpretation of the CCC data.

A second case-study is presented in Subsection 4.3.3, which was obtained from the New South Wales Breast Cancer Institute, Australia.[3] This second case-study has been developed from the casuistic point of view to "allow clinicians, health professionals, and members of the public to examine and understand some of the controversial and difficult aspects of breast cancer management". The data from the second case-study are therefore more detailed than the patient data from the first case-study and are more suitable for an investigation of critiquing from a clinical viewpoint.

4.3.2. *Case study 1: ductal carcinoma in situ*

The steps of critiquing on one specific patient derived from the data, and subsequently interpreted by medical experts, is illustrated here. The diagnosis and treatment is summarised in Fig. 3. It can be said that this is a rather typical patient as it is a patient with one of the most frequent diagnoses in the data records. The following property describes the treatment sequence that our example patient has undergone.

> *"For a patient with diagnosis Ductal Carcinoma In Situ (DCIS), the following sequence of states is possible: the treatment starts, then axillary staging by sentinel node is activated, after which breast conserving therapy is activated".*

To specify and then verify that breast conserving therapy (denoted bct) can take place after axillary staging by sentinel node procedure (denoted asbSN), the following CTL formula is used:

$$\mathbf{EF}(\mathrm{DCIS} \wedge \mathbf{EX}\,\mathbf{EF}(\mathrm{asbSN} \wedge \mathbf{EX}\,\mathbf{EF}\,\mathrm{bct})) \tag{10}$$

A more strict formula could be obtained by assuming that the diagnosis DCIS, holds up to the moment of breast con-

[1] CBO: Richtlijn Behandeling van het mammacarcinoom, van Zuiden, 2002.

[2] Breast cancer model obtained from `http://www.protocure.org`

[3] `http://www.bci.org.au/medical/caseindex.htm`

Medical condition: 79 years-old woman. Lesion of right breast: carcinoma in-situ with size between 1 and 2 cm. Two lymph nodes investigated and none positive.
Treatment: sentinel node biopsy followed by breast-conserving surgery without axillary clearance.

Fig. 3. Description of patient in conjunction with the prescribed treatment.

serving therapy. However, this property stated above turns out to be false as it is, i.e., this treatment is non-compliant with respect to the guideline. In other words, according to the model of the guideline describing the treatment of DCIS, the sequence of actions performed by the doctor is incorrect for this patient. It could also be explained by the fact that, according to the model, at least one of the two actions in patient treatment should not be started, or they should be started in a different sequence. To identify this inconsistency, we reduce the actions that are being performed. If we reduce the sequence to only one action, then both actions are found possible, as shown by the following property (corresponding to the case when only 'bct' is activated as part of the DCIS treatment):

$$\mathbf{EF}(\mathrm{DCIS} \wedge \mathbf{EX}\,\mathbf{EF}\,\mathrm{bct}) \tag{11}$$

The new conclusion is that under these circumstances the two actions cannot be activated in this sequence, or the ordering should be reversed.

In the experiment on the seven fairly prototypical patient-cases that can be found in the Dutch CCC dataset, some deviation was found between the guideline and each of the seven prototypical cases. Interestingly, for three of these, some differences could indeed be explained by looking at the new 2004 revision of the guideline.

4.3.3. *Case study 2: infiltrating ductal carcinoma*

For the second case study we have more elaborate information available. It concerns a patient who is a female with a lump in the 3 o'clock position of the right breast and a second lump just above this. No palpable axillary nodes or other abnormalities were found. The mammography revealed no focal mass, grouped microcalcifications, or anatomic distortion. Finally, the histopathology showed two lesions: both infiltrating duct carcinoma, 20mm in size, and with similar morphology. The sentinel nodes were mapped using lymphoscintigraphy and a biopsy was taken of a right axillary lymph node and an internal mammary node (the sentinel node procedure). In the right axillary lymph node, no malignancy was found. However, in the internal mammary node, metastatic carcinoma was identified. The treatment consisted of a total mastectomy of the right breast with immediate reconstruction. The axilla was treated by means of an axillary clearance and re-section of two further internal mammary nodes at higher levels (these were sampled partly because of the original pathology finding and partly because of ready access to the IMC).

The vocabulary of the guideline does not include the term 'infiltrating ductal carcinoma', but rather discusses 'operable invasive breast cancer' (OIBC). According to the guideline, operable invasive breast cancer is defined as T1-2 N0-1 M0, i.e., a tumour smaller than 5cm, with maximally one lymph node positive, and no distant metastasis. On basis of information provided by the diagnostic tests, the patient can be considered part of this patient group. Each of the three interventions (sentinel node procedure, mastectomy, and axillary clearance) can be mapped to terms found in the guideline. This can be done with reasonable confidence, however, some details have to be ignored such as the re-section of the internal mammary nodes as part of the axillary clearance, as this part of the treatment is not mentioned in the guideline. With respect to the order between interventions, it is only clear that the sentinel node procedure (asbSN) is performed before the other two interventions.

The treatment can again be critiqued using a CTL proof obligation, but as some of the information is missing here, we illustrate critiquing using modular model checking. The proof obligation is then described by $[\varphi]M\langle\bot\rangle$ where

$$\varphi = \mathbf{X}\,\mathrm{OIBC} \wedge \mathbf{F}\,(\mathrm{asbSN} \wedge \mathbf{X}(\mathbf{F}\,\text{axillary-clearance} \wedge \mathbf{F}\,\text{mastectomy})) \tag{12}$$

The proof obligation $[\varphi]M\langle\bot\rangle$ is true, showing that this combination of interventions is *not* possible (cf. Subsection 4.1.2). The reason for this can be further analysed by removing one of the order constraints yielding

$$\varphi' = \mathbf{X}\,\mathrm{OIBC} \wedge \mathbf{F}\,\mathrm{asbSN} \wedge \mathbf{F}\,\text{axillary-clearance} \wedge \mathbf{F}\,\text{mastectomy} \tag{13}$$

As $[\varphi']M\langle\bot\rangle$ is true, the formula φ' is further weakened by removing one of the interventions from the conjunct. This results in three new proof obligations, showing that the guideline model does not contain a trace with both a sentinel node procedure and axillary clearance for this patient, while all other combinations appear to be possible. Thus, the conclusion is that the combination of actions that are being prescribed is non-compliant with respect to the guideline.

5. Critiquing based on satisfaction sets

In the previous section we describe a critiquing method that uses a model checker for checking whether the patient description and treatment are consistent with the formal model of the guideline. In this section, we describe a method that uses a model checker to determine all the possible patient findings for which a certain property holds in the formal model of the guideline. This compiled information from the guideline is then used for critiquing the patient data.

5.1. *Introduction to the method*

This indirect method for critiquing is based on exploiting generated counterexamples from a model checker. We have specifically designed CTL formulas such that we obtain useful counterexamples. These counterexamples enable us to construct sets of parameter values for which a property *Prop* holds. We call such a set a satisfaction set for *Prop.*

Definition 1 (Satisfaction set) *The* satisfaction set *for a property of a plan with respect to the formal model of a guideline is the set of parameter-value combinations, representing a set of findings, and denoted by parameter = value, for which the property holds in the model.*

The satisfaction set for a property *Prop* with respect to a certain model has the following form:

$$\begin{aligned} \{ \ & \langle p_1 = v_{11}, p_2 = v_{21}, p_3 = v_{31}, \ldots, p_n = v_{n1} \rangle, \\ & \langle p_1 = v_{12}, p_2 = v_{22}, p_3 = v_{32}, \ldots, p_n = v_{n2} \rangle, \\ & \qquad \vdots \\ & \langle p_1 = v_{1s}, p_2 = v_{2s}, p_3 = v_{3s}, \ldots, p_n = v_{ns} \rangle \ \} \end{aligned}$$

Where each p_i is a parameter and each v_{ij} the values for those parameters, with n the number of parameters and s the number of parameter combinations. Each of these combinations represents a particular value constellation for which property *Prop* holds. We use this information for critiquing the patient data: if the patient parameters do not correspond with one of the parameter combinations from a satisfaction set for property *Prop* then this property *Prop* does not hold for the patient data. Examples of such a property are whether a treatment can be started, or whether a treatment is completed.

Fig. 4 shows the critiquing method based on satisfaction sets. We start by constructing a CTL formula for a certain property that we check against the model of the guideline with a model checker. This formula will either be true or generate a counterexample which we use to extend the CTL formula. We continue extending the CTL formula using counterexamples until the formula is true. This true formula codifies the satisfaction set for the property that the CTL formula was built for. This satisfaction set is than compared against the patient data to see whether that patient data is compliant with the guideline or not and if not, what the reason is for being non-compliant. Notice that the construction of the satisfaction sets can be done off-line, independent of the actual patient records.

Next, we show how to generate satisfaction sets to critique patient data.

5.2. *Satisfaction sets for completed treatments*

We will look at generating and using satisfaction sets in detail for a specific property, namely compliance of a treatment. As we discussed in Section 2, there are several different interpretations of the concept ‘compliance of a treatment’. The CTL formulae that express these different interpretations have in common that they all describe constraints on the state of a plan. These constraints effectively represent a set of SMV states in which the plan is in a particular state. We can construct a satisfaction set for any property that describes a set of SMV states, so it is possible for each interpretation of compliance to construct a satisfaction set.

For the rest of this section, we use the interpretation that treatment compliance means that a treatment is successfully completed, using the formalisation in Formula (1), which is true in the state just before a plan becomes completed.

The satisfaction sets for completed treatments enable us to find many types of discrepancies between the patient data and the guideline model.

Now we will describe how to generate satisfaction sets for a completed treatment and how to use these sets for critiquing the patient data against the guideline model such that we are able to identify several types of non-compliance.

Generating satisfaction sets for completed treatments

Our goal is to find all possible combinations of findings for which Formula (1) holds for treatment t. We start by checking if all parameters p_i are always unknown if plan t can be completed:

$$\mathbf{AG}\,(t \rightarrow (p_1 = \mathit{unknown} \wedge \ldots \wedge p_n = \mathit{unknown})) \tag{14}$$

When one uses all the available parameters from the guideline in this formula and the following formulae, the satisfaction set is guaranteed to cover all possible parameter value combinations. The analysis can be restricted by taking a subset of all the parameters. The value *unknown* is the initial value for any parameter before it is assigned a value from the guideline. This happens whenever a plan is executed, so if a plan can be completed at all, this formula is false and will generate a counterexample that shows a parameter value combination for which the plan can be completed.

We can use the parameter value combinations from the counterexample to extend the initial **AG** formula for further construction of the satisfaction set. The following formula then has to be checked:

$$\begin{aligned} \mathbf{AG}\,(t \rightarrow\ & ((p_1 = \mathit{unknown} \wedge \ldots \wedge p_n = \mathit{unknown}) \vee \\ & (p_1 = v_{11} \wedge p_2 = v_{21} \wedge \ldots \wedge p_n = v_{n1})) \end{aligned} \tag{15}$$

Notice that the new disjunct contains the parameter values of the first counter example. If this formula is true, it means that treatment t can only be completed if $p_1 = v_{11} \wedge p_2 = v_{21} \wedge \ldots \wedge p_n = v_{n1}$. If there are more parameter values under which t can be completed, this formula will again be false. The counterexample will then provide us with another combination of parameter values

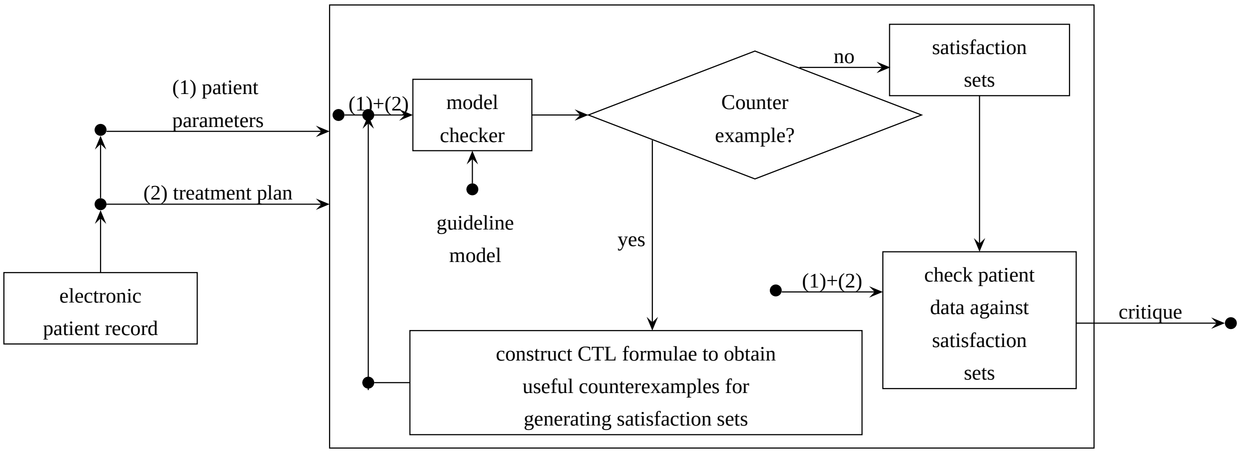

Fig. 4. Critiquing approach using model checking by exploiting counter examples.

for which t can be completed. We add this combination to Formula (15) as an additional disjunct in the disjunction. This extended formula has to be checked again. We repeat these steps until the formula is true. This process will always terminate, as there are a finite number of parameter combinations, and the number of possible values for any parameter is finite.

Using satisfaction sets for completed treatments

We can use satisfaction sets for completed treatments to find discrepancies between the patient record and the guideline. We compare the satisfaction set for each treatment in the patient record to the patient parameters of the patient record. If the patient parameters are not identical to one of the parameter value combinations from a satisfaction set of a particular treatment, that treatment can not be completed with those patient parameters. This means that either those patient parameters are wrong (non-compliance type F2) or the action should not be part of the treatment (non-compliance type T2).

If two satisfaction sets do not have two identical parameter value combinations, then there is no parameter value combination for which both plans can be completed.[4] In other words, the plans are mutually exclusive. As this can only be the case if at least one of the two plans can not be completed, we only check for mutually exclusive plans if a type T2/F2 discrepancy has been found.

We call a parameter irrelevant for the completion of a plan if there is no value for that parameter for which the plan can not be completed. In other words, the parameter does not influence the completion of that plan. We can find irrelevant parameters with the following method: suppose we want to determine whether parameter p_1 is irrelevant for plan t. We have to check whether for each possible value for p_1, there is a parameter value combination in which p_1 has a different value, but all the other parameters have identical values. For example, if p can have values a, b, or c, p_1 is irrelevant if the following parameter value combinations are part of the satisfaction set of t:

$$\langle p_1 = a, p_2 = v_2, \ldots, p_n = v_n\rangle$$
$$\langle p_1 = b, p_2 = v_2, \ldots, p_n = v_n\rangle$$
$$\langle p_1 = c, p_2 = v_2, \ldots, p_n = v_n\rangle$$

If a parameter is irrelevant for each plan from the treatment of a patient record, that parameter is irrelevant for that patient description and therefore could be omitted from it. This is exactly non-compliance type F4.

All parameters that are not irrelevant for the completion of a plan are relevant. The relevant parameters of a treatment description in the patient record should be part of the patient parameters. We simply check whether all relevant parameters are mentioned in the patient description. These missing relevant parameters are non-compliance type F1.

Notice that we consider the parameters as irrelevant independent of other parameters. This has the consequence that we can not detect that parameter p_1 or p_2 is irrelevant. We consider them both as irrelevant, p_1 is irrelevant because it is not strictly needed because of p_2, vice versa p_2 is irrelevant because of p_1. This has consequences for the non-compliances type F4 and F1 that we identify. Concerning “F4 containing irrelevant parameters” our method is unsound, but complete. This means that we give possibly to many parameters that are irrelevant. Using this analysis, the physician does not need to look to all the patient parameters to check if these are irrelevant for the treatment, but only to the indicated part of these patient parameters. However it could be still the case that one of these selected parameters is not irrelevant. Concerning “F1 missing relevant parameters” our method is sound, but incomplete. This means that it is possible that there are more missing relevant parameters then we are giving. At least the one we give are indeed necessary for compliance between patient data and guideline.

[4] Excluding the parameter value combination in which each parameter is initialized with ‘unknown’.

Note that we can not find any information regarding correct action sequences (non-compliance type T3, F3) in the patient data with these satisfaction sets. For this we would need satisfaction sets that are constructed based on formulae that take into account the sequence of plans instead of only the completion of a plan.

5.3. *Case study*

In this case study we use fictive patient data to show how critiquing using satisfaction sets works. The patient data is as follows:

Patient description
Diagnosis: locally advanced breast cancer
Intensive treatment contraindications: no
Hormone receptor status: negative
Treatment trace
aht, neo, rsm, lrm [a]

[a] (adjuvant hormonal therapy, neoadjuvant chemotherapy, radical surgery, locoregional radiotherapy)

We critique this patient data against a guideline for treating breast cancer. This patient data is inconsistent with the breast cancer guideline, as the guideline prescribes that the action ‘aht’ should only be executed when the tumour is hormone receptor-positive, which is not the case in this patient data. We will now demonstrate how to generate satisfaction sets with this patient data and what we can derive from these sets.

5.3.1. *Generating satisfaction sets*

We have to generate a treatment completed satisfaction set for each of the four actions from the patient record. We will show this process only for the action ‘aht’ as this process is identical for the other actions. We start using the following formula:

$$\begin{aligned}
&\mathbf{AG}\,(t \rightarrow (\\
&\quad (\mathit{diagnosis} = \mathit{unknown} \wedge\\
&\quad \mathit{hormone\_receptor\_status} = \mathit{unknown} \wedge\\
&\quad \mathit{intensive\_treatment\_contraindications} = \mathit{unknown}))
\end{aligned}$$

When model checking this formula we receive the following counterexample:

Values of parameters
Diagnosis = locally_advanced_BC
intensive_treatment_contra_indications = yes
hormone_receptor_status = positive

We use this information to check whether the initial parameter value set is already a satisfaction set or whether it should be extended. This is done with the following formula:

$$\begin{aligned}
&\mathbf{AG}\,(t \rightarrow (\\
&\quad (\mathit{diagnosis} = \mathit{unknown} \wedge\\
&\quad \mathit{hormone\_receptor\_status} = \mathit{unknown} \wedge\\
&\quad \mathit{intensive\_treatment\_contraindications} = \mathit{unknown}) \vee\\
&\quad (\mathit{diagnosis} = \mathit{locally\_advanced\_BC} \wedge\\
&\quad \mathit{hormone\_receptor\_status} = \mathit{positive} \wedge\\
&\quad \mathit{intensive\_treatment\_contraindications} = \mathit{yes}))
\end{aligned}$$

Notice that the second disjunct is exactly the counter example found in the previous step. This formula, however, is again false and gives another counterexample:

Values of parameters
Diagnosis = locally_advanced_BC
intensive_treatment_contraindications = no
hormone_receptor_status = positive

So we extend the formula with these values and check again whether this extended set of parameter values is a satisfaction set:

$$\begin{aligned}
&\mathbf{AG}\,(t \rightarrow (\\
&\quad (\mathit{diagnosis} = \mathit{unknown} \wedge\\
&\quad \mathit{hormone\_receptor\_status} = \mathit{unknown} \wedge\\
&\quad \mathit{intensive\_treatment\_contraindications} = \mathit{unknown}) \vee\\
&\quad (\mathit{diagnosis} = \mathit{locally\_advanced\_BC} \wedge\\
&\quad \mathit{hormone\_receptor\_status} = \mathit{positive} \wedge\\
&\quad \mathit{intensive\_treatment\_contraindications} = \mathit{yes})) \vee\\
&\quad (\mathit{diagnosis} = \mathit{locally\_advanced\_BC} \wedge\\
&\quad \mathit{hormone\_receptor\_status} = \mathit{positive} \wedge\\
&\quad \mathit{intensive\_treatment\_contraindications} = \mathit{no}))
\end{aligned}$$

This formula is true, meaning that we have a satisfaction set for the action ‘aht’. We repeat this process for the other three actions, which results in the satisfaction sets shown in Fig. 5.

5.3.2. *Using satisfaction sets*

With these satisfaction sets we can perform a number of checks in order to detect various types of non-compliance between guideline and patient data.

First, we determine if any of the actions in the treatment trace can not be completed. We do this by checking whether the patient parameters are identical with a parameter value combination from the satisfaction set of each action. There is one action for which the patient parameters do not fall

Satisfaction set of action aht:

{ ⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = yes ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = no ⟩ }

Satisfaction set of action lrm:

{ ⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = negative, itc = yes ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = yes ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = negative, itc = no ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = no ⟩ }

Satisfaction set of action neo:

{ ⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = negative, itc = no ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = no ⟩ }

Satisfaction set of action rsm:

{ ⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = negative, itc = no ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = negative, itc = yes ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = yes ⟩,
⟨ diagnosis = locally_advanced_BC, hormone_receptor_status = positive, itc = no ⟩ }

Fig. 5. Complete satisfaction sets ('itc' is intensive_treatment_contraindications).

within its satisfaction set, namely 'aht'. Plan 'aht' can only be completed if 'hormone_receptor_status' has a positive value, as all its parameter value combination have a positive value for 'hormone_receptor_status', but the patient data has a negative value for this parameter. This means that 'aht' can not be completed for the patient parameters and that the reason for this is that the hormone receptor status in the patient data do not match the required values for 'aht'. We report this discrepancy to the medical expert. This also means that the patient data does not comply with the guideline. We continue the analysis to see whether the patient data has other discrepancies.

Plan	Irrelevant parameters
aht	intensive_treatment_contraindications
lrm	intensive_treatment_contraindications, hormone_receptor_status
neo	hormone_receptor_status
rsm	intensive_treatment_contraindications, hormone_receptor_status

Fig. 6. Irrelevant parameters for each plan.

Second, we check if the patient data contains irrelevant parameters. The irrelevant parameters for each plan are shown in Fig. 6. These are the parameters that in the satisfaction sets have all the possible values they can have. As there are no parameters that are irrelevant for all plans from the treatment trace, there is no irrelevant parameter in the patient data. (No type F4 non-compliance).

Third, we check if the patient data misses a relevant parameter. As all the parameters from the satisfaction sets are in the patient data, this is not the case. (No type F1 non-compliance).

Fourth, we determine if there are any mutually exclusive actions in the treatment trace. As 'aht' can not be completed for the given patient parameters, we have to check whether 'aht' is mutually exclusive with any of the other three actions. We do this by checking for each pair of actions from the treatment trace whether they have two matching parameter value combinations. This is the case for all the satisfaction sets of all the plans and therefore there are no mutually exclusive actions in this treatment description (No special case of type T2).

We report that the patient data does not comply with the guideline. In addition, we make an intersection of all the satisfaction sets of the treatment trace so we can tell the medical expert for what values this treatment trace would be compliant. We also report this to the medical expert.

An automatically generated report of the analysis given to a medical expert could look like Fig. 7.

This report includes an T2 non-compliance (there is an action in the treatment trace that should not be executed for these parameters) and its F2 variant (hormone_receptor_status must have a positive value, whereas the patient data has a negative value).

Patient data
Diagnosis: locally advanced breast cancer
Intensive treatment contraindications: no
Hormone receptor status: negative
Treatment trace: aht, lrm, neo, rsm

Results of the analysis of this patient data

- The patient data does not comply with the guideline. The parameter values for which the treatment trace would comply are:
  - diagnosis: must have value locally_advanced_BC (patient data has same value)
  - intensive_treatment_contraindications: must have value no (patient data has same value)
  - hormone_receptor_status: must have value positive (patient data has the value negative)
- There is a non-completed action in the treatment trace. According to the guideline, the treatment 'aht' should not be part of the treatment trace given the patient parameters. The reason for this is that 'aht' should only be part of treatments for which hormone_receptor_status has value positive, while in this patient data the hormone_receptor_status parameter has value negative.
- There are no irrelevant parameters in this patient data.
- There are no missing relevant parameters in this patient data.
- No mutually exclusive actions have been found in the treatment trace.

Fig. 7. Critiquing report for medical expert.

This case study shows that in principle the use of satisfaction sets can be helpful with critiquing patient data against a guideline.

6. Related work

The use of the term *critiquing* to describe a system that criticises the solution provided by a human can be attributed to Miller [15], who developed his ATTENDING system for critiquing anaesthesia management. The earliest systems relied heavily on user interaction, resulting in rejection [9]. To avoid this problem, we followed in this paper the approach originally proposed by van der Lei [13], of gathering all information from computer-stored medical records. Although critiquing has first been used for evaluating medical treatments, since then it has been applied to a wide variety of problems such as engineering design, decision making, word processing, knowledge base acquisition, and software engineering [22].

At the end of the 1990s, when several guideline representation languages were introduced, adopting critiquing to clinical practice guidelines became a topic of interest. Two major limitations, however, make 'classical' critiquing systems hardly acceptable in practice: (1) these systems are unable to cope with deviations from the underlying model, and (2) these systems are unable to cope with the question why a physician was performing a particular action [23].

Several authors [20, 21, 1, 2], have therefore advocated to address these limitations by assessing the compliance of a physician's intentions with the intentions of the guideline. Although formal guideline representation languages such as Asbru, EON, GUIDE, and Pro*forma* are able to express intentions [18], in [14], a case study showed that intentions of clinical practice guidelines are often left implicit. Following an intention based critiquing approach, the ability to recognize the plan of the physician is therefore considered to be an indispensable prerequisite to the performance of plan critiquing [21]. However, such an approach would need a large body of knowledge about the effects of interventions on clinical parameters and knowledge of domain-independent and domain specific guideline revision strategies. For example, [23] makes use of the pharmacotherapeutic compass to retrieve intentions of drug administration.

The intention based approaches can in principle be integrated with our approach, but have not been pursued here in order to focus on critiquing in terms of model checking of a temporal logic representation, which only recently appeared in literature [24, 5]. None of the approaches mentioned above use a formal logical representation. One reason for this, is that current guideline representation languages often lack a logical semantics. Recently, progress has been made in providing a logic semantic to some guideline representation languages [4, 3, 16].

In this paper we used Asbru for specifying the guideline model and used LTL and CTL for verifying properties about the critiquing process. Such properties typically have a qualitative nature, e.g., they deal with order between treatments. Quantitative properties such as 'stop the use of blood thinners 14 days prior to surgery' could also be expressed in this framework albeit with more difficulty. Even when a continuous time model is adopted, subsets of LTL and CTL are often used to analyse timed systems, as for example in the Uppaal model checker [12, 10]. More complex expressions such as 'check glucose values four times per day' may require more expressive logics, e.g., logics with fixpoint operators, such as the well-known μ-calculus [7]. However, formulas in such logics are difficult to comprehend and increase the computational complexity.

7. Discussion and conclusions

7.1. *Comparison*

In this paper, we have described two methods for critiquing: critiquing by directly verifying properties of the guideline model and critiquing using an indirect method. We summarise some differences between the two methods.

On-line vs off-line costs. The indirect method consists of two computational tasks, namely the generation of satisfaction sets and critiquing on the basis of these satisfaction sets. Its advantage compared to the direct method is that the first task can be done "off-line", in advance, independent of any specific patient data. The resulting "compilation" of the guideline traces can then be reused across many different patient records.

Representation of the model. The explicit state space of the model of the guideline is very large. For example, the model that was used for this paper contains hundreds of variables and a compact representation of its state space using binary decision diagrams [6] ranges between 2 and 4 million nodes [5] depending on the properties of interest. This state space, however, can be succinctly represented in the SMV specification language (in this case, the size of the specification is 300KB including comments).

It would be very unwise to represent the satisfaction sets explicitly. The breast cancer model used for this paper consists of 50 actions and about 25 binary parameters, which implies a (theoretical) limit of $2^{25} \approx 3.35 \times 10^7$ elements in the satisfaction set, for each action. Satisfaction sets, however, can also be represented using binary decision diagrams, as they represent a Boolean function over the patient findings. As most actions depend only on a few of these findings, e.g., the majority of the findings is used for only one action (so they are irrelevant to all other actions), BDDs can provide a compact representation of the satisfaction sets.

Time complexity of "on-line critiquing". As the various non-compliances have a somewhat different complexity in each method, we simplify matters by not considering iterations needed for finding some of the compliances. Then the complexity of the direct method is determined by the complexity of model checking (i.e., searching through the whole model), whereas for indirect critiquing we have to search in the satisfaction sets. It follows from the fact that each element of the satisfaction set is a state in the search space (as they are derived from counterexamples), that the indirect method must be at least as efficient. In practice, the guideline has many more states due to the overhead of the semantics of Asbru which defines the state of each action. Satisfaction sets abstract from these structures as well as to various other features (e.g., repetition).

Handling the non-compliance types. In order to illustrate both approaches and various non-compliances, the case study of Section 4 focused on the detection of non-compliance of type T3 ("non-compliant order of actions"), and T2 ("non-compliant actions"). The indirect method from Section 5 focused on whether an action is completed, and this enables us to identify the F1, F2, F4, T1, and T2 types of non-compliance. As we have argued, both methods are capable of answering questions regarding any non-compliance. However, in case there are iterations needed to answer questions regarding the compliance of a certain treatment or parameter, the difference in running time between searching through the whole model versus satisfaction sets becomes even more important.

7.2. *Formal assumptions and medical relevance*

In this paper we have applied a formal, mathematical framework (model-checking) for modelling a real-life task (critiquing based on medical guidelines). As always, the use of a mathematical framework to model a real-life task involves making a number of assumptions that are demanded by the mathematical framework. Such assumptions serve to simplify some aspects of the real-life task in order for the mathematical framework to be applicable. Hence, such assumptions determine the medical relevance of the mathematical model: if the assumptions imposed by the mathematical framework are too strong, the mathematical model is not realistic, and the results are no longer relevant to actual medical practice.

The framework presented in this paper has also by necessity made a number of such simplifying assumptions. In this section we will enumerate a number of these assumptions, and we will discuss the extent to which these assumptions affect the medical relevance of the results of this paper. This discussion will then also serve as a guide to future work, where one might try to relax some of the more restrictive assumptions.

Closed world assumption on abnormality of patient data. The traditional definition of the "closed world assumption" (CWA, [19]), is that when a fact is not explicitly stated as true, it can assumed to be false. This assumption is true in most database systems, and has been inherited in many knowledge representation systems.

For medical records, the situation is somewhat more complex. For lab-data, the traditional CWA is mostly applicable: data which have been obtained are listed, and if a data-value is not given, we can assume it to be false. For other data (e.g., data obtained by anamnesis), a modified version seems more appropriate: parameters with normal values are often not recorded in the patient record. Hence, the CWA becomes that "if a parameter is not explicitly listed with an abnormal value, we can assume that the parameter has a normal value."

In this paper, our non-compliance categories T1 and F1 rely on making the classical CWA: if data-values are not given, they can be assumed to be false. It would be interesting to extend our framework with notions of abnormality of data-values in order to apply the medically more relevant version of this assumption.

Data persistence. As stated in section 3.2, we assume that parameters do not dynamically change during the execution of the plan: patient findings are valid before any interventions have been performed. Of course, in clinical practice, the patient's status can change due to unexpected reasons which are exogenous with respect to the execution of actions in the guideline. Hence, relaxing this assumption would increase the medical relevance of our results. One possibility is to refine the temporal formulas in the direct method to take into account the dynamic nature of the parameters. However, this will make the formulas much more complex.

Instantaneous actions. Many temporal logics make the assumption of instantaneous actions. This assumption states that an action started in some state is finished in the next state, and that all the effects of the actions are directly visible in that next state. As shown by formula (1) we make a much weaker assumption. This formula states that if an action is active in state α it should be *possible* for the action to complete in *one* of the successor states, but our assumption does not require that the completion *must* happen in *all* successor states. Hence, this formulation allows for durative actions, since it allows for a (possibly) long chain of states where the action is not completed.

Mandatory actions. In this paper, we have made the rather strict assumption that if the patient's data satisfy the preconditions of a mandatory action in the guideline, then that action must be executed. Of course, this is not always the case in clinical practice. For example, if unexpected complications occur that are risky for the patient then the standard guideline execution is suspended in favour of a treatment of the complications, resulting in not executing an action that was listed as mandatory. Nevertheless, the non-execution of a mandatory action is still an important pointer to a *potential* non-compliance and hence deserves highlighting.

Data granularity. Throughout this work, we have assumed that the data in the patient record is described using the same terminology and at the same level of granularity as used in the guideline. Unfortunately, this is far from being the case in clinical practice. Hence, any clinically deployable method for guideline-based critiquing must include methods for bridging this gap, e.g., by deploying abstraction [21] and vocabulary mapping [17]. Alternatively, some modern guidelines are now beginning to *prescribe* the data-collection that must be done in the patient record when a patient is treated under that guideline. This then prevents the gap between record and guideline from occurring.

7.3. *Future work*

Degree of automation. Even though the steps in the analysis of the case studies was done manually, it is not difficult to see how to automate this process since the temporal formulas could be generated mechanically. A more challenging question is how to use the result of this process for the construction of a human readable critique. In evidence-based guidelines, explanation and references are often provided, however, formal models of guidelines often abstract from this information making it difficult to provide elaborate information to the practitioner.

Efficient generation of satisfaction sets. A more detailed inspection of the structure of the formalised guideline model is likely to yield information about the (in)dependence of findings, and the irrelevance of certain findings to certain actions. This information can be used to be more economic in the generation or use of satisfaction sets, since not all combinatorially possible satisfaction sets will have to be actually generated.

Categorizing other critiquing methods using the compliance types. In the comparison above, we have labelled each of our critiquing methods with the non-compliance types that they can detect. It would be interesting to extend this exercise to other critiquing methods that have been reported in the literature [22]. This would provide a better insight in the relative strengths and weaknesses of the various methods.

Closer coordination between guidelines and patient records. Our empirical work on actual patient records under the breast cancer guideline has revealed a significant gap between the vocabulary used to describe actions and findings in the guideline and the vocabulary used in patient records. Future deployment of critiquing systems would certainly benefit from an alignment of the vocabularies used in guidelines and patient records. Ontology-based techniques would seem to be appropriate here [17].

Similarly, patient records are notoriously incomplete. Many actions are not time-stamped, and even the recording of an action is often ambiguous as to whether this concerns the start-date or the completion-date of an action, and which success-status the action was completed.

Clinical validation on the usefulness of the non-compliances. An important question concerning the non-compliance types is their clinical usefulness: are some, all, or none of the given non-compliance types of actual interest in a clinical setting, and have we perhaps failed to identify other types which are of clinical interest.

Extend the satisfaction set method to deal with missing mandatory actions. Currently, the satisfaction-set method from Section 5 is unable to detect missing mandatory actions (non-compliance type T4). The method would need to be extended to deal with this.

Generating and using sequence satisfaction sets. Satisfaction sets for sequence of plans contain the parameter value combinations for which a certain plan precedes another plan. Constructing these satisfaction sets is mainly the same as with satisfaction sets for completed treatment, as their construction is also based on exploited counterexamples. A different initial CTL formula is used, but the same way of extending this formula is used. With this type of satisfaction set we can determine non-compliance of the types T3 and F3.

Making the step from non-compliance detection to repair actions. Of course, detecting a non-compliance is only the first step of a longer process. Once a non-compliance is detected, the question arises how this non-compliance should be "repaired". Is the non-compliance due to a mistake in the treatment of the patient? Or due to a mistake (or incompleteness) in the formulation of the guideline? Or was the original guideline correct, and was an error introduced in the formal modelling process?

7.4. *Conclusions*

The main conclusion of this work is that it is, in principle, possible to use model checking on formalised models in order to critique medical guidelines against patient data. We have shown how critiquing can be characterised in temporal logic and have applied this to a case study on the treatment of breast cancer. We have also used a model checker for determining all the possible patient parameter values for which a certain property holds in the formal model of the guideline. This compiled information from the guideline is then used for critiquing the patient data.

Model checking provides additional value to a simulation-based critiquing of an operational version of the guideline. Such critiquing based on running the operational guideline model through an interpreter only checks the consistency of a patient record against a single trace through the guideline (namely, the one chosen by the interpreter), while model checking compares the patient record against all possible traces through the guideline.

A strong aspect of model checking is a high degree of automation as compared for example to theorem proving, making it suitable for deployment in a critiquing system. The fully automated nature of model checking, however, also brings a weakness with it. In principle, model checking only *detects* inconsistencies, but does not contribute to the interpretation of the inconsistency. In general, model checking can construct a counter-example illustrating the inconsistency, which is often a very good guide towards tracing its source. However, this only works when model checking global properties, i.e., properties dealing with all possible treatment paths, while critiquing the existence of a single treatment makes it impossible for the model checker to generate a counter-example. In this paper, we have proposed some general strategies to deal with this (repeated experiments with weaker specifications by relaxing order constraints and by removing actions).

A general conclusion with respect to the breast cancer case study that can be drawn is that a closer correspondence is needed between the processes of guideline construction and data-collection. In fact, this is currently already being partially implemented by the Dutch Institute of Health care Improvement: newly constructed guidelines are currently being equipped with a data-collection dictionary, which will ensure the correspondence between collected data and guideline terminology.

References

[1] A. Advani, K. Lo, and Y. Shahar. Intention-based critiquing of guideline-oriented medical care. In *Proceedings of AMIA Annual Symposium*, pages 483–487, 1998.

[2] A. Advani, Y. Shahar, and M.A. Musen. Medical quality assessment by scoring adherence to guideline intentions. *J Am Med Inform Assoc.*, 9:S9–S97, 2002.

[3] M. Alberti, A. Ciampolini, F. Chesani, M. Gavanelli, P. Mello, M. Montali, S. Storari, and P. Torroni. Protocol specification and verification using computational logic. In *WOA*, pages 184–192, 2005.

[4] M. Balser, C. Duelli, and W. Reif. Formal semantics of Asbru - an overview. In *Proceedings of the International Conference on Integrated Design and Process Technology*, Passadena, June 2002. Society for Design and Process Science.

[5] S. Bäumler, M. Balser, A. Dunets, W. Reif, and J. Schmitt. Verification of medical guidelines by model checking – a case study. In A. Valmari, editor, *Proceedings of 13th International SPIN Workshop on Model Checking of Software*, volume 3925 of *LNCS*, pages 219–233. Springer-Verlag, 2006.

[6] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986.

[7] E.M. Clarke, O. Grumberg, and A.D. Peled. *Model Checking.* The MIT Press, Cambridge, Massachusetts, London, England, 2001.

[8] A.S. Gertner. *Critiquing: effective decision support in time-critical domains.* PhD thesis, Dept. of Computer & Information Science, University of Pennsylvania, 1995.

[9] A.S. Gertner. Plan recognition and evaluation for online critiquing. *User Modeling and User-Adapted Interaction*, 7(2):107–140, 1997.

[10] T.A. Henzinger. Symbolic model checking for realtime systems. *Information and Computation*, 111:193–?244, 1994.

[11] O. Kupferman and M.Y. Vardi. Modular model checking. *Lecture Notes in Computer Science*, 1536:381–401, 1998.

[12] K.G. Larsen, P. Peterson, and Y. Wang. UPPAAL in a nuttshell. *Journal of Software Tools for Technology Transfer*, 1(1-2):134–152, 1997.

[13] J. van der Lei. *Critiquing based on Computer-Stored Medical Records.* PhD thesis, Erasmus University, 1991.

[14] M. Marcos, G. Berger, F. van Harmelen, A. ten Teije, H. Roomans, and S. Miksch. Using critiquing for improving medical protocols: Harder than it seems. In *8th European Conference on Artificial Intelligence in Medicine*, pages 431–441, 2001.

[15] P. Miller. *A critiquing approach to Expert Computer Advice: ATTENDING.* Pittman Press, London, 1984.

[16] N. Mulyar, M. Pesic, W.M.P. van der Aalst, and M. Peleg. Towards the flexibility in clinical guideline modelling languages. In *5th International Conference on BPM - Workshop BPM in Healthcare*, pages 17–28, 2007.

[17] Chintan Patel, James J. Cimino, Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Li Ma, Edith Schonberg, and Kavitha Srinivas. Matching patient records to clinical trials using ontologies. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 816–829. Springer, 2007.

[18] M. Peleg, S. Tu, J. Bury, P. Ciccares, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, and M. Stefanelli. Comparing computer interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*, 10:52–68, 2004.

[19] R. Reiter. On closed world databases. In Gallaire and Minker, editors, *Logic and Databases*, pages 55–76. Plenum, 1978.

[20] Y. Shahar, S. Miksch, and P. Johnson. A task-specific ontology for the application and critiquing of time oriented clinical guidelines. In *Proceedings of the sixth Conference on Artificial Intelligence in Medicine in Europe*, pages 51–61, 1997.

[21] Y. Shahar, S. Miksch, and P. Johnson. The Asgaard project: A task-specific framework for the application and critiquing of time-orientied clinical guidelines. *AIME*, 14:29–51, 1998.

[22] B.G. Silverman. Survey of expert critiquing systems: Practical and theoretical frontiers. *Communications of the ACM*, 35(4):106–127, 1992.

[23] R. Sips, L. Braun, and N. Roos. Applying formal medical guidelines for critiquing. In *ECAI 2006 WS – AI techniques in healthcare: evidence based guidelines and protocols*, 2006.

[24] P. Terenziani, L. Giodano, A. Bottrighi, S. Montani, and L. Donzella. SPIN model checking for the verification of clinical guideline. In *ECAI 2006 WS – AI techniques in healthcare: evidence-baded guidelines and protocols*, 2006.