

# Enhancing Reaction Systems with Guards for Analysing Comorbidity Treatment Strategies\*

Juliana Bowles<sup>1,2</sup>, Linda Brodo<sup>3</sup>, Roberto Bruni<sup>4</sup>, Moreno Falaschi<sup>5</sup>,  
Roberta Gori<sup>4</sup>, and Paolo Milazzo<sup>4</sup>

<sup>1</sup> School of Computer Science, University of St Andrews, St Andrews, UK.

Email: [jkfb@st-andrews.ac.uk](mailto:jkfb@st-andrews.ac.uk)

<sup>2</sup> Software Competence Centre Hagenberg Softwarepark 32a, Hagenberg, Austria.

<sup>3</sup> Dipartimento di Scienze Economiche e Aziendali, Università di Sassari,

Via Muroni 25, Sassari, Italy. Email: [brodo@uniss.it](mailto:brodo@uniss.it)

<sup>4</sup> Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, Pisa, Italy.

Emails: {roberto.bruni, roberta.gori, paolo.milazzo}@unipi.it

<sup>5</sup> Department of Information Engineering and Mathematics, University of Siena,

Via Roma 56, Siena, Italy. Email: [moreno.falaschi@unisi.it](mailto:moreno.falaschi@unisi.it)

**Abstract.** The coexistence of multiple medical conditions in an individual presents a complex challenge in healthcare. This research aims to enhance the analysis of comorbidity treatment plans by capturing nuanced interactions between different medical conditions and treatment interventions. Reaction systems provide a formal framework for modelling and analysing systems in which the dynamics is driven by promotion/inhibition mechanisms and external intervention of context processes. This study explores the integration of guards into context processes to increase the expressiveness of the formalism in order to analyse treatment plans for comorbidities. Through the analysis of combined clinical guidelines for atrial fibrillation and hypertension, we demonstrate the applicability and utility of this approach in optimizing patient care and improving health outcomes in the context of complex medical scenarios.

**Keywords:** Reaction systems, Clinical guidelines, SOS semantics.

## 1 Introduction

Over time, it is common for people to develop long-term chronic conditions, such as diabetes, hypertension, cardiovascular diseases, chronic kidney disease, cancer, chronic obstructive pulmonary disease, among many others. Patients diagnosed with such conditions are inevitably required to take medications as part

---

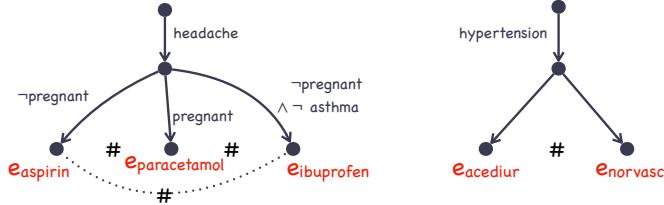
\* Research supported by the Next Generation EU project PNRR ECS00000017 “THE - Tuscany Health Ecosystem” (Spoke 3 - CUP B63C22000680007, and Spoke 6 - CUP I53C22000780001), the Next Generation EU projects MEDICA (PRIN 2022, CUP\_B53D23013170006 CUP\_I53D23003720006), DELICE (PRIN PNRR 2022, P20223T2MF), RAP (PRIN PNRR 2022, P2022HXNSC), the INdAM GNCS project CUP\_E53C22001930001, and the Austrian Science Fund (FWF) project Meitner M 3338-N.

of their treatment and/or to alleviate individual symptoms. Clinical guidelines were devised as evidence-based care plans to help manage the treatment of a wide range of health conditions [16,22]. They detail the essential steps followed when caring for patients with a specific clinical problem and play an important role in improving the healthcare provision for people with long-term conditions. These guidelines include recommendations for (group of) medications to be given at different stages of the treatment plan as well as alternatives, and are revised regularly. However, for patients with two or more chronic conditions (aka *multimorbidities*), these guidelines are insufficient and it is hard to have an oversight of the problems underlying following several care plans at the same time. Patients with multimorbidities follow several treatment plans simultaneously, which easily leads to *polypharmacy* (i.e., prescribing 5 or more medications) without clear guidance on how best to prioritise recommendations [17]. The risk of medication harm is exacerbated, that is, it is possible for patients to take medications that lead to adverse drug reactions, or for particular combinations of drugs to be less effective if administered at the same time.

Our study is motivated by the need to search for treatment paths that respect the preferences of patients with multimorbidities. This includes *looking ahead* for paths that will lead to fewer complications later on. Making the right choice as early as possible in the therapy (given the conditions of patients and the effects of their progression) can avoid worse outcomes in later stages.

In this paper, we use formal methods to model guidelines, different patients profiles and known adverse drug reactions. Our analysis intends to help doctors choose between alternative treatment options by pointing out the possible risks in choosing a particular therapy (also in the long term). As a side effect, our analysis may also identify missing conditions that should be considered in future revised guidelines (e.g., certain groups of drugs offered in one treatment should not be offered to patients with a particular comorbidity due to risky interactions between medications). To this aim, we chose to use Reaction Systems (RSs) [15,7], a computational framework that describes the dynamics of a system in terms of a set of reactions, each with some reactants, inhibitors and products. The *facilitation* and *inhibition* mechanisms inherent to RSs allow us to formalise medical guidelines in a modular way. The execution of a RS starts when an external environment, known as *context*, provides an initial set of entities, all the enabled reactions are fired and the union of their products are released. This step is iterative: next, the context provides another set of entities that are combined with the products from the previous step to fire enabled reactions again.

To model the scenario at hand, we must enhance the expressiveness of contexts so that they can supply different entities depending on those that are already present. We adopt the process algebra version of RS [9], which is easier to extend as already shown in [10,12], and we add a guarded prefix operator. Exploiting the same promotion/inhibition mechanisms of reactions, guarded contexts can check some properties of the actual state before providing new entities. Using this new functionality, we are able to encode guidelines as contexts that are driven by patient profiles, their conditions, and their current medications.



**Fig. 1.** Treatments for headache (on the left) and hypertension (on the right).

Our main result is the systematic encoding of guidelines, patient profiles and adverse drugs reactions in the enhanced version of R<sub>S</sub>s. Our encoding relies on an unambiguous representation of medical guidelines. In particular, we adopt the event structure modelling of therapies introduced in [3], paired with a tabular representation of adverse drug reactions which are the result of known side effects of drug combinations. The synthesised model is then used to derive a graph that describes, for each patient profile, all possible evolutions of patient therapies in accordance to clinical guidelines. Finally, we can derive a table that illustrates, for each patient profile, the choices that could lead into a critical medical situation.

**Plan of the paper.** In Section 2 we introduce clinical guidelines with a toy example. Section 3 introduces the basics of R<sub>S</sub>s, that we extend with guards in Section 4. Section 5 outlines our modelling approach step by step. Section 6 presents our main clinical case study. Section 7 shows the efficacy of our approach on the case study. Section 8 presents related work and draws some conclusion.

## 2 Concise Guidelines Description

Medical guidelines are provided as long and complex textual documents. The analysis of guidelines with computational methods requires them to be expressed in a more schematic representation, suitable for computer processing. In [3], a representation is proposed based on the formalism of *event structures* paired with a table of adverse drug reactions for known side effects of combined usage of drugs. Turning a real medical guideline into such a representation and maintaining it updated over time requires human work and curation, that would be feasible if effective computational analysis methods were available. As we aim to propose one such analysis methods, we adopt the approach proposed in [3] and assume guidelines to be already available in such a concise form. For the guidelines used in this paper as case studies, the event structure representations were derived manually from the real and updated guideline text.

To illustrate the formal description of guidelines, we introduce two toy examples that will also be used to motivate the introduction of guarded contexts in R<sub>S</sub>s (cf. Section 4) as well as to exemplify our general methodology in Section 5.

The first therapy describes the prescription process of a painkiller for migraine, that can be either aspirin or paracetamol or ibuprofen. The second therapy represents the prescription process of two different drugs for hypertension:

acediur contains the active ingredients captopril and hydrochlorothiazide (a diuretic drug) while norvasc is a medicine containing the active substance amiodipine besylate. The two therapies are depicted as event structures in Fig. 1. Each node represents an event and each directed arc an (immediate) causal dependency (the source precedes the target). The symbol # denotes (immediate) conflicting events. Events not related by causality or conflict are called concurrent. Arc labels represent guards that must be satisfied to take that path. Node labels show the drug prescriptions associated with that event.

For example, in the case of migraine, the top event represents the start of the treatment that leads to the prescription of one drug among aspirin or paracetamol or ibuprofen (one conflicting event for each choice). Guards such as pregnant,  $\neg$ pregnant and  $(\neg\text{pregnant} \wedge \neg\text{asthma})$  expose some factors that must be considered when choosing the right painkiller: patients with asthma are generally told to avoid ibuprofen while pregnant patient must take paracetamol only.

When combining the two therapies, some adversarial effects may arise. Indeed, taking aspirin and diuretics (that are contained in acediur) may lead to a (moderate) risk of renal toxicity. We represent potential drug conflicts as entries in a companion table called, Adverse Drug Reaction (ADR) table.

Adverse Drug Reactions			
Medication	Medication	Reaction	Side effects & risks include
aspirin	acediur	moderate	renal toxicity

Guarded events structures and ADR tables form the input of our method that aims to model, analyse and possibly improve medical guidelines within the RS framework. Patient profiles, patient conditions, medications and conflicts between adverse drugs will be modelled directly by reactions. Since reactions are deterministic, guarded event structures must necessarily be modelled by RS contexts. Since different therapies can be guided by guards on patient conditions, such as in the example of the painkiller above, we need to extend RS contexts in such a way that the choice to take can be driven by the current state.

### 3 Background on Reaction Systems

First, we briefly account for the set theoretic definition of Reaction Systems (RSs) [15,7]. Then, we focus on their process algebraic version [9].

**RS basics.** Let  $S$  be a finite set of *entities*. A *reaction* in  $S$  is a triple  $a = (R, I, P)$ , where  $R, I, P \subseteq S$  are finite, non-empty sets and  $R \cap I = \emptyset$ . The sets  $R, I, P$  are the sets of *reactants*, *inhibitors*, and *products*, respectively. Without loss of generality, we admit the use of empty sets of reactants and inhibitors.

A Reaction System is a pair  $\mathcal{A} = (S, A)$ , where  $S$  is the set of entities, and  $A$  is a finite set of reactions in  $S$ . Given the current state  $W \subseteq S$ , a reaction  $a = (R, I, P)$  is enabled in  $W$  if its *enabling predicate*  $en_a(W) \triangleq R \subseteq W \wedge I \cap W = \emptyset$  is satisfied. The *result* of the reaction  $a$  on the current state  $W$  is defined as  $res_a(W) \triangleq P$  if  $en_a(W)$ , and  $res_a(W) \triangleq \emptyset$  otherwise. The result of all reactions  $A$  on the current state  $W$ , is  $res_A(W) \triangleq \cup_{a \in A} res_a(W)$ .

The no-permanency principle of RSs dictates that entities disappear if not sustained by some reaction. Thus, the current state  $W = D \cup C$  is determined by the result  $D$  of all reactions on the previous state, together with additional entities  $C$  provided by the *context*. The dynamics of a RS is formalised in terms of *interactive processes*. They are pairs  $\pi = (\gamma, \delta)$ , where  $\gamma \triangleq \{C_i\}_{i \in [0, n]}$  is the *context sequence* and  $\delta = \{D_i\}_{i \in [0, n]}$  is the *result sequence*, s.t.:  $D_0 = \emptyset$ ; and  $D_{i+1} = \text{res}_A(D_i \cup C_i)$  for any  $i \in [0, n - 1]$ . The sequence  $\gamma$  represents external interventions, while the sequence  $\delta$  is entirely determined by  $\gamma$  and  $A$ . We call  $\tau \triangleq \{W_i\}_{i \in [0, n]}$  the *state sequence*, with  $W_i \triangleq C_i \cup D_i$  for any  $i \in [0, n]$ .

*Example 1.* Let  $S \triangleq \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ ,  $A \triangleq (S, A)$  and  $\gamma \triangleq \{\mathbf{a}, \mathbf{b}\}, \{\mathbf{a}\}, \emptyset$ , where  $A \triangleq \{a_1\}$  contains a unique reaction  $a_1 \triangleq (\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}\}, \{\mathbf{b}, \mathbf{c}\})$ . Then, the result sequence is  $\delta \triangleq \emptyset, \{\mathbf{b}, \mathbf{c}\}, \emptyset$  and the state sequence is  $\tau = \{\mathbf{a}, \mathbf{b}\}, \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}, \emptyset$ , because, e.g.,  $\text{res}_A(\{\mathbf{a}, \mathbf{b}\}) = \{\mathbf{b}, \mathbf{c}\}$  and  $\text{res}_A(\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}) = \emptyset$ .

**Process algebraic RSs.** Inspired by Plotkin's Structural Operational Semantics (SOS) approach [20] and process algebras such as CCS [19], we derive a Labelled Transition System (LTS) semantics for RSs by means of inductive inference rules. LTS states are terms of an algebra, each transition defines a computation step of the RS and its label records the entities involved in that step. Following [9], we also admit nondeterministic and recursively defined contexts.

**Definition 1 (RS processes).** RS processes are defined by the grammar below:

$$\mathsf{P} := [\mathsf{M}] \quad \mathsf{M} := (R, I, P) \mid D \mid \mathsf{K} \mid \mathsf{M}|\mathsf{M} \quad \mathsf{K} ::= \mathbf{0} \mid C.\mathsf{K} \mid \mathsf{K} + \mathsf{K} \mid X$$

where  $R$ ,  $I$ ,  $P$ ,  $C$ , and  $D$  are sets of entities (with  $P \neq \emptyset$  and  $R \cap I = \emptyset$ ) and  $X$  is a context identifier drawn from a family of (recursive) definitions  $\Delta \triangleq \{X_j = \mathsf{K}_j\}_{j \in J}$ , called the environment.

A RS process  $\mathsf{P}$  embeds a *mixture* process  $\mathsf{M}$  obtained as the parallel composition of some reactions  $(R, I, P)$ , some available entities  $D$  (if any), and some context  $\mathsf{K}$ . We write  $\prod_{i \in I} \mathsf{M}_i$  for the parallel composition of all  $\mathsf{M}_i$  with  $i \in I$ .

A context process  $\mathsf{K}$  is either: the nil context  $\mathbf{0}$  that stops the computation; the prefixed context  $C.\mathsf{K}$  that makes the entities  $C$  available to the reactions, and then will behave as  $\mathsf{K}$  at the next step; the non-deterministic choice  $\mathsf{K}_1 + \mathsf{K}_2$  that can behave as either  $\mathsf{K}_1$  or  $\mathsf{K}_2$ ; the context identifier  $X$  that behaves as  $\mathsf{K}$  for  $X = \mathsf{K} \in \Delta$ . For example, if  $A = \{\mathbf{a}\}.A + \emptyset.A \in \Delta$  and  $B = \{\mathbf{b}\}.B + \emptyset.B \in \Delta$ , then the context  $A|B$  can offer any combination of  $\mathbf{a}$  and  $\mathbf{b}$  at each step.

We say that  $\mathsf{P}$  and  $\mathsf{P}'$  are structurally equivalent, written  $\mathsf{P} \equiv \mathsf{P}'$ , when they denote the same term up to the laws of Abelian monoids (unit, associativity and commutativity) for parallel composition  $\cdot|.$ , with  $\emptyset$  as the unit, and the laws of idempotent Abelian monoids for choice  $\cdot + \cdot$ , with  $\mathbf{0}$  as the unit. We also assume  $D_1|D_2 \equiv D_1 \cup D_2$  for any  $D_1, D_2 \subseteq S$ .

**Definition 2 (RSs as RS processes).** Let  $A = (S, A)$  be a RS, and  $\pi = (\gamma, \delta)$  an interactive process, with  $\gamma = \{C_i\}_{i \in [0, n]}$  and  $\delta = \{D_i\}_{i \in [0, n]}$ . For any step  $i \in [0, n]$ , the corresponding RS process  $[\![\mathcal{A}, \pi]\!]_i$  is defined as follows:

$$[\![\mathcal{A}, \pi]\!]_i \triangleq [ \prod_{a \in A} a \mid D_i \mid C_i.C_{i+1} \cdots .C_n.\mathbf{0} ]$$

$$\begin{array}{c}
\frac{}{D \xrightarrow{\langle(D,\emptyset) \triangleright \emptyset, \emptyset, \emptyset\rangle} \emptyset} \text{(Ent)} \quad \frac{}{C.K \xrightarrow{\langle(\emptyset,C) \triangleright \emptyset, \emptyset, \emptyset\rangle} K} \text{(Cxt)} \\
\frac{K_1 \xrightarrow{\ell} K'_1}{K_1 + K_2 \xrightarrow{\ell} K'_1} \text{(Suml)} \quad \frac{K_2 \xrightarrow{\ell} K'_2}{K_1 + K_2 \xrightarrow{\ell} K'_2} \text{(Sumr)} \quad \frac{X = K \in \Delta \quad K \xrightarrow{\ell} K'}{X \xrightarrow{\ell} K'} \text{(Rec)} \\
\frac{}{(R,I,P) \xrightarrow{\langle(\emptyset,\emptyset) \triangleright R,I,P\rangle} (R,I,P) | P} \text{(Pro)} \quad \frac{J \subseteq I \quad Q \subseteq R \quad J \cup Q \neq \emptyset}{(R,I,P) \xrightarrow{\langle(\emptyset,\emptyset) \triangleright J,Q,\emptyset\rangle} (R,I,P)} \text{(Inh)} \\
\frac{M_1 \xrightarrow{\ell_1} M'_1 \quad M_2 \xrightarrow{\ell_2} M'_2 \quad \ell_1 \frown \ell_2}{M_1 | M_2 \xrightarrow{\ell_1 \cup \ell_2} M'_1 | M'_2} \text{(Par)} \quad \frac{M \xrightarrow{\langle(D,C) \triangleright R,I,P\rangle} M' \quad R \subseteq D \cup C}{[M] \xrightarrow{\langle(D,C) \triangleright R,I,P\rangle} [M']} \text{(Sys)}
\end{array}$$

**Fig. 2.** SOS semantics of the RS processes.

The SOS semantics of RS processes is defined by the SOS rules in Fig. 2.

A transition label  $\ell$ , written  $\langle(D, C) \triangleright R, I, P\rangle$ , records: the sets  $D$  of entities currently in the system; the set  $C$  of entities provided by the context; the set  $R$  of entities whose presence justifies some reactions; the set  $I$  of entities whose absence justifies some reactions; and the set  $P$  of reaction products. The SOS rules guarantee that if  $P \xrightarrow{\langle(D,C) \triangleright R,I,P\rangle} P'$  it holds  $en_{(R,I,P)}(D \cup C)$ .

The rule (Ent) records the set of current entities  $D$ . By rule (Cxt), a prefixed context process  $C.K$  makes available the entities in  $C$  and then reduces to  $K$ . Rules (Suml) and (Sumr) select a move of either the left or the right context, resp., discarding the other process. By rule (Rec), a context identifier  $X$  behaves according to its defining process  $K$ . The rule (Pro) assumes the reaction  $(R, I, P)$  is enabled: it records its reactants, inhibitors, and products in the label, and leaves the reaction available at the next step, together with its products  $P$ . The rule (Inh) records in the label the reasons why the reaction  $(R, I, P)$  should not be executed: possibly some inhibiting entities ( $J \subseteq I$ ) are present or some reactants ( $Q \subseteq R$ ) are missing, with  $J \cup Q \neq \emptyset$ , as at least one cause is needed. The rule (Par) puts two processes in parallel by pooling their labels and joining all labels components. We write  $\ell_1 \cup \ell_2$  for the component-wise union of labels:

$$\bigcup_{i=1,2} \langle(D_i, C_i) \triangleright R_i, I_i, P_i\rangle \triangleq \langle(D_1 \cup D_2, C_1 \cup C_2) \triangleright R_1 \cup R_2, I_1 \cup I_2, P_1 \cup P_2\rangle.$$

The sanity check  $\ell_1 \frown \ell_2$  is required to guarantee that labels of reactants and inhibitors are consistent (see definition below):

$$\langle(D_1, C_1) \triangleright R_1, I_1, P_1\rangle \frown \langle(D_2, C_2) \triangleright R_2, I_2, P_2\rangle \triangleq (\bigcup_{i=1,2} D_i \cup C_i \cup R_i) \cap (I_1 \cup I_2) = \emptyset.$$

Finally, the rule (Sys) checks that all the needed reactants are available in the system ( $R \subseteq D \cup C$ ). Checking the absence of inhibitors ( $(D \cup C) \cap I = \emptyset$ ) is not necessary, thanks to the sanity check in rule (Par). Notably, the SOS semantics matches the set-theoretic dynamics of RSs (as made precise by [9, Th. 19]).

*Example 2.* The encoding of the RS in Example 1, is as follows:

$$[\![\mathcal{A}, \pi]\!]_0 \triangleq [ (ab, c, bc) \mid \emptyset \mid \{a, b\}. \{a\}. \emptyset. \mathbf{0} ] \equiv [ a_1 \mid \{a, b\}. \{a\}. \emptyset. \mathbf{0} ].$$

Then, from the SOS rules in Fig. 2, we derive, e.g.:

$$[\![\mathcal{A}, \pi]\!]_0 \xrightarrow{\langle(\emptyset, ab) \triangleright ab, c, bc\rangle} [a_1 | \{b, c\} | \{a\}. \emptyset. \mathbf{0}] \xrightarrow{\langle(bc, a) \triangleright c, \emptyset, \emptyset\rangle} [a_1 | \emptyset. \mathbf{0}] \xrightarrow{\langle(\emptyset, \emptyset) \triangleright \emptyset, ab, \emptyset\rangle} [a_1 | \mathbf{0}].$$

$$\begin{array}{c}
\frac{}{D \xrightarrow{\langle\langle D \triangleright \emptyset, \emptyset, \emptyset \rangle\rangle \triangleright \emptyset, \emptyset, \emptyset} \emptyset} \text{(Ent)} \\
\frac{}{(R, I, C).K \xrightarrow{\langle\langle \emptyset \triangleright R, I, C \rangle\rangle \triangleright \emptyset, \emptyset, \emptyset} K} \text{(Cxt)} \\
\frac{}{(R, I, P) \xrightarrow{\langle\langle \emptyset \triangleright \emptyset, \emptyset, \emptyset \rangle\rangle \triangleright R, I, P} (R, I, P)|P} \text{(Pro)} \\
\frac{J \subseteq I \quad Q \subseteq R \quad J \cup Q \neq \emptyset}{(R, I, P) \xrightarrow{\langle\langle \emptyset \triangleright R, I, C \rangle\rangle \triangleright J, Q, \emptyset} (R, I, P)} \text{(Inh)} \\
\frac{M \xrightarrow{\langle\langle D \triangleright R', I', C \rangle\rangle \triangleright R, I, P} M' \quad R' \subseteq D \quad R \subseteq D \cup C}{[M] \xrightarrow{\langle\langle D \triangleright R', I', C \rangle\rangle \triangleright R, I, P} [M']} \text{(Sys)}
\end{array}$$

**Fig. 3.** Updated rules to handle guarded contexts (rules (*Suml*), (*Sumr*), (*Rec*) and (*Par*) are left unchanged from Fig. 2 and thus omitted).

## 4 Reaction Systems with Guards

We enrich the expressiveness of context processes by introducing the possibility to provide some entities *whenever the current state satisfy certain conditions*. To this aim, we introduce guarded prefixes  $(R, I, C).K$ .

**Definition 3 (Guarded contexts).** We update Definition 1 by letting:

$$K ::= \mathbf{0} \mid (R, I, C).K \mid K + K \mid X$$

where  $R, I, C \subseteq S$  are possibly empty sets of entities such that  $R \cap I = \emptyset$  and  $X$  is a context identifier drawn from a predefined environment  $\Delta$ .

The similarity with the syntax of reactions is intended: the key difference is that in  $(R, I, C).K$  the presence of reactants  $R$  and the absence of inhibitors  $I$  is checked w.r.t. the set of current entities  $D$ . More importantly, the products  $C$  are made available immediately from the context, not at the next step. We admit  $R$  and  $I$  to be possibly empty, and abbreviate  $(\emptyset, \emptyset, C).K$  as  $C.K$ . Note that a conditional prefixed process that is not enabled behaves as the  $\mathbf{0}$  process.

The SOS rules are updated as shown in Fig. 3. To handle guarded contexts we need a more sophisticated kind of labels: we write  $\ell$  for a tuple of the form

$$\langle\langle D \triangleright R', I', C \rangle\rangle \triangleright R, I, P$$

whose novel component  $\langle\langle D \triangleright R', I', C \rangle\rangle$  records the available entities  $D$  together with those provided by the guarded contexts, assuming all entities in  $R'$  are present and those in  $I'$  are absent. The SOS rules guarantee that whenever  $P \xrightarrow{\langle\langle D \triangleright R', I', C \rangle\rangle \triangleright R, I, P} P'$  it holds that  $en_{(R', I', C)}(D)$  and  $en_{(R, I, P)}(D \cup C)$ . Consequently, we overload the notation  $\ell_1 \frown \ell_2$  and  $\ell_1 \cup \ell_2$  by letting:

$$\begin{aligned}
& \langle\langle D_1 \triangleright R'_1, I'_1, C_1 \rangle\rangle \triangleright R_1, I_1, P_1 \frown \langle\langle D_2 \triangleright R'_2, I'_2, C_2 \rangle\rangle \triangleright R_2, I_2, P_2 \\
& \triangleq (\bigcup_{i=1,2} D_i \cup R'_i) \cap (I'_1 \cup I'_2) = \emptyset \wedge (\bigcup_{i=1,2} D_i \cup C_i \cup R_i) \cap (I_1 \cup I_2) = \emptyset \\
& \langle\langle D_1 \triangleright R'_1, I'_1, C_1 \rangle\rangle \triangleright R_1, I_1, P_1 \cup \langle\langle D_2 \triangleright R'_2, I'_2, C_2 \rangle\rangle \triangleright R_2, I_2, P_2 \\
& \triangleq \langle\langle D_1 \cup D_2 \triangleright R'_1 \cup R'_2, I'_1 \cup I'_2, C_1 \cup C_2 \rangle\rangle \triangleright R_1 \cup R_2, I_1 \cup I_2, P_1 \cup P_2
\end{aligned}$$

*Example 3.* The context  $X = (\{\mathbf{a}\}, \emptyset, \{\mathbf{b}\}).X + (\{\mathbf{b}\}, \emptyset, \{\mathbf{a}\}).X + (\emptyset, \{\mathbf{a}, \mathbf{b}\}, \emptyset).X$  ensures that whenever  $\mathbf{a}$  or  $\mathbf{b}$  are present in the current state, then both  $\mathbf{a}$  and  $\mathbf{b}$  are available as reactants. The last option  $(\emptyset, \{\mathbf{a}, \mathbf{b}\}, \emptyset).X$  handles the case where the first two guards fail: if omitted, when both  $\mathbf{a}$  and  $\mathbf{b}$  are absent the context  $X$  would behave as  $\mathbf{0}$ , blocking the computation.

## 5 RS Models for Clinical Guidelines

In this section, we outline our methodology for the modelling of clinical guidelines using RSs. Given the concise formal description of therapies in terms of guarded event structures and ADR tables, we distil a RS process model. To illustrate our construction, we take the toy guidelines from Section 2 as a running example.

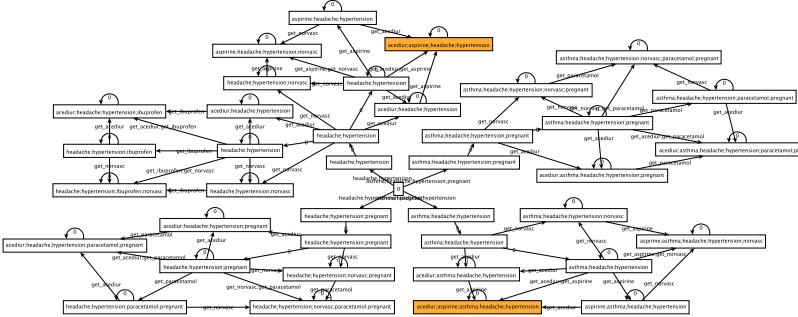
*The void process.* We always include an auxiliary process  $\text{Void} = \emptyset.\text{Void}$  that recursively provides the empty set of entities at any step. It will eventually lead to some attractor, so to model the long terms effect of the therapy.

*Patient profile.* Therapies are often based on the health conditions of the patient. In the toy example, the patient profile is determined by the conditions that trigger the treatment (headache, hypertension) and by the conditions that appear in the arc labels (pregnant, asthma). We call them *features*. The patient profile is determined by any combination of features. In the experimentation, there will be exactly one context  $K_f = \emptyset.\text{Void} + \{f\}.\text{Void}$  for each feature  $f$ . When they are composed in parallel, their synchronous execution will account for any possible combination of features. In the example, the patient profile will be modelled by the context process:  $K_{\text{headache}} \mid K_{\text{hypertension}} \mid K_{\text{pregnant}} \mid K_{\text{asthma}}$ . Alternatively, specific profiles can be investigated by removing one alternative from each  $K_f$ . Once the profile is determined by the context, it is preserved during the rest of the computation by reactions of the form  $(\{f\}, \emptyset, \{f\})$ , one for each feature. In the example, we have, e.g.,  $(\{\text{headache}\}, \emptyset, \{\text{headache}\}) \mid \dots \mid (\{\text{asthma}\}, \emptyset, \{\text{asthma}\})$ .

*Drugs.* For each drug  $d$  that appears in the therapies, we consider three corresponding entities  $\text{get}_d$ ,  $\text{stop}_d$  and  $d$ : the first represents the prescription of  $d$  by the doctor, the second the removal of  $d$  from the current treatment and the third the intake of the drug by the patient. Typically, drugs are grouped into classes  $c$  and we introduce two more entities  $c$  and  $\text{stop}_c$  for each drug class  $c$  when needed. Entities  $\text{get}_d$  and  $\text{stop}_d$  will be provided by the context that models the guideline. For each drug,  $d$  there will be the following reactions:  $(\{\text{get}_d\}, \{\text{stop}_d, \text{stop}_c\}, \{d, c\})$  modelling the intake of the drug  $d$  as for doctor prescription, and  $(\{d\}, \{\text{stop}_d, \text{stop}_c\}, \{d, c\})$  modelling the prosecution of the therapy. Note the presence of  $\text{stop}_d$  and  $\text{stop}_c$  as inhibitors. In the toy example we have five drugs (aspirin, paracetamol, ibuprofen, acediur and norvasc), but omit their classes for simplicity. Hence, we introduce reactions such as

$$(\{\text{get\_aspirin}\}, \{\text{stop\_aspirin}\}, \{\text{aspirin}\}) \mid (\{\text{aspirin}\}, \{\text{stop\_aspirin}\}, \{\text{aspirin}\}) \mid \dots$$

*ADR and alerts.* A generic entry of the ADR table consists of two drugs  $d_1$  and  $d_2$  and some side effects  $m$ . For simplicity, we assume that side effects are divided in just three categories: major, moderate and minor. To each entry, we thus introduce a reaction  $(\{d_1, d_2\}, \emptyset, \{m\})$ . In the toy example we have the reaction  $(\text{aspirin acediur}, \emptyset, \text{moderate})$ . Moreover, other kinds of alert can be raised in an analogous way, for example, since a drug can cure different symptoms, a particular alert will be raised in the case study of Section 7 when two different drugs of the same class are prescribed as a result of combined therapies.



**Fig. 4.** LTS for (toy) headache/hypertension therapies.

*Therapies.* We assume that therapies are described as guarded event structures, which we translate to context processes. Roughly, we introduce an identifier  $E_e$  for each event  $e$  and define it as a sum of processes, one for each outgoing arc of  $e$ . If some guard is attached to the arc, then the corresponding alternative is also guarded. When some drug prescription for  $d$  is present in a node, then the corresponding choice produces  $\text{get}_d$ . Similarly, if the therapy requires stopping some drug  $d$ , the entity  $\text{stop}_d$  is produced. Concurrent events are activated separately. For example, the first guideline of our toy example is modelled as

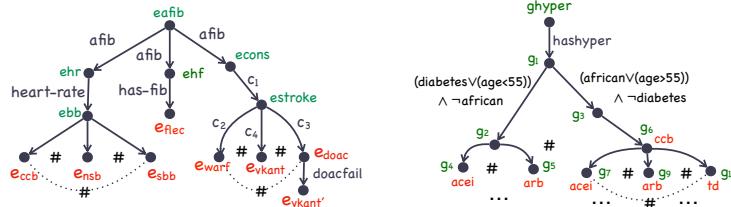
$$E_{\text{headache}} \triangleq (\emptyset, \{\text{pregnant}\}, \emptyset).E_a + (\{\text{pregnant}\}, \emptyset, \emptyset).E_p + (\emptyset, \{\text{pregnant}, \text{asthma}\}, \emptyset).E_i$$

where we let  $E_a \triangleq \{\text{get}_\text{aspirin}\}.\text{Void}$ ,  $E_p \triangleq \{\text{get}_\text{paracetamol}\}.\text{Void}$ , etc.

*Timing issues and asynchronous execution.* The computational model of RSs is based on a fully synchronous semantics. This implies some degree of unmotivated synchronization in the simulation of concurrent therapies. E.g., in two concurrent therapies each consisting of two drugs to be administered in sequence, the first drug of the first therapy would be administered together with the first drug of the second therapy, and the same would hold for the two second drugs. This would happen in the simulation, regardless of the actual timing of the involved drugs. In order to abstract away from all timing issues, we design an asynchronous model of therapies. The idea is that passing from our drug to another is the result of a prescription by a doctor who evaluates the status of the patient and decides how to proceed on the basis of the guidelines. This can happen at any time. Enforcing asynchronicity in the model allows the analysis to provide general results that hold independently of the specific prescription timings. To introduce asynchrony in the simulation of a therapy, we can slightly modify the above definitions to account for arbitrary time lapses before the actual drug prescription, e.g., we can set  $E_a \triangleq \emptyset.E_a + \{\text{get}_\text{aspirin}\}.\text{Void}$ , etc. In what follows, we will assume therapies to be modelled in an asynchronous way.

*The whole model.* The above fragments are assembled in parallel to form a single RS process. In this paper, we will exploit the following features of our RS toolkit:

1. automatic LTS generation in different formats (.dot, .graphml, .pdf, etc.) with customizable appearance: typically we label nodes with features, drugs



**Fig. 5.** Line of treatment for patients with atrial fibrillation (left) and first steps of drug treatment for hypertension (right).

- and side effects and arcs with all entities supplied by the context; moreover, by using different colours for side effects (red for major, orange for moderate) we highlight the risks that patients may undergo (see Fig. 4);
2. automatic synthesis of patient profiles that are more at risk;
  3. dynamic guideline revision: by refining guarded context to prevent severe effects for specific patients, we can readily check the efficacy of the changes.

Our framework has high flexibility. By tuning a few parameters, we can experiment with the same medical therapies under different circumstances (selected patient profiles, synchronous or asynchronous treatments, etc.).

## 6 A Clinical Case Study: AFib and Hypertension

Consider a diagnosis of atrial fibrillation (AFib), a heart condition which causes the heart rate to be irregular and often abnormally too fast. Treatment for AFib normally involves medications to *control the heart rate*, *restore normal heart rhythm* and *reduce the risk of stroke*. To control the heart rate patients can take: either standard beta blockers (SBB) (e.g., atenolol and bisoprolol), non-standard ones (NSBB) (e.g. carvedilol and propranolol) or a (rate limiting) calcium channel blocker (CCB) (e.g., verapamil and diltiazem). AFib patients may also have to take flecainide to restore the normal rhythm. Moreover, patients following one of the two previous therapies are at risk of blood clots forming in the heart chambers, which can cause a stroke. Hence, they can take either warfarin or a direct-acting anticoagulant (DOAC) (e.g., dabigatran and apixaban).

The guideline for AFib is shown in Fig. 5 (left) where a patient with AFib, if needs it, should be treated for medications to control heart rate, to restore the heart rate and to reduce the risk of stroke. The guard  $c_1$  tests if the patient has at least a heart rate or a fibrillation problem and if he agrees to take a drug to reduce the stroke risks. In that case, the choice of the drug class is made on the basis of complex guards: if a patient is younger than 75 and does not have other comorbidities such diabetes or hypertension (guard  $c_2$ ) the doctor should prescribe warfarin. Otherwise, the suggested therapy includes a DOAC drug unless the patient is intolerant or there are evidences that that therapy will not work (guard  $c_3$ ). In that case (guard  $doacfail$ ) the cure is switched to vitamin K antagonist, the same for intolerance to DOAC drugs (guard  $c_4$ ).

Adverse Drug Reactions			
Medication (class)	Medication	Reaction	Side effects & risks include
apixaban (doac)	any ccb	moderate	bleeding issues
dabigatran (doac)	diltiazem (ccb)	moderate	bleeding issues
dabigatran (doac)	verapamil (ccb)	major	anaemia, bleeding issues
dabigatran (doac)	carvedilol (nsbb)	moderate	bleeding issues
warfarin	benazepril (acei)	minor	
warfarin	any td	minor	
warfarin	propranolol(nsbb)	minor	
flecainide	any ccb	major	chest pain, coma, ...
flecainide	any sbb	moderate	fainting, palpitations
flecainide	any nsbb	moderate	fainting, palpitations

**Table 2.** Interactions between drugs used for treating AFib and Hypertension.

Now consider that the patient has also been diagnosed with hypertension<sup>6</sup>. whose first treatment steps are shown in Fig. 5 (right). Patients without type 2 diabetes but younger than 55 follow the same treatment as patients with diabetes type 2. In this case, the treatment choice consists of an ACEI or an ARB. Patients older than 55 or black African or African-Caribbean family origin (any age) are offered a different first treatment with *ccb*. Further steps, (not shown) due to intensification, involve adding one by one new medications not chosen before.

Drugs with clear adverse reactions are shown in Table 2.

## 7 Modelling and Analysis of the Clinical Case Study

We now apply the toolkit outlined in Section 5 to the case study in Section 6. For each of the two guidelines in Section 6, we detail the entities, reactions and guarded contexts representing patient features, drugs and therapies. The whole RS model, consisting of 123 reactions and 50 context process declarations and written in the syntax of the analysis tool BioReSolve [2], is available online [14].

**AFib Guideline.** *Patient profile.* By observing the guards in the event structure in Fig. 5, it emerges that patient features of interest are: *afib* (has to enter the AFib guideline), *heart\_rate* (has heart rate problem), *has\_fib* (has fibrillation), *consensus\_acei* (gives consensus to treatment for stroke risk reduction), *over75* (is above 75 years old), *diabetes* (has diabetes), *hyper* (has hypertension), *doac\_int* (is intolerant to DOAC). Context processes for feature setting and reactions for their maintenance are as explained in Section 5. Moreover, a guard in the right branch of the event structure from state *e\_doac* is based on the result of the DOAC therapy prescribed in such a state. Hence, we include in the RS model two more entities, *doac\_ok* and *doac\_fail*, representing the success and the failure of such a therapy, respectively. These two entities are comparable to patient features (they contribute to describe the health status of the patient), but they are not set at the beginning of the analysis. As soon as the DOAC therapy is prescribed for the first time, either *doac\_ok* or *doac\_fail* is non-deterministically produced by the RS model in order to explore both possibilities in the analysis.

<sup>6</sup> See details in the NICE guideline NG136 at [www.nice.org.uk](http://www.nice.org.uk)

**Drugs.** Drugs administered in the AFib guideline together with their classes are modelled by the following entities: atenolol and bisoprolol (class sbb), carvedilol and propranolol (class nsbb), verapamil and diltiazem (class ccb), warfarin (no class specified), dabigatran and apixaban (class doac), vkant (vitamin K antagonist). Reactions for drug prescription and interruption are as explained in Section 5.

**Therapies.** Since the AFib guideline (see Fig. 5) activates three concurrent events ehr, ehf and econs, the initial context process includes three parallel processes eafib1, eafib2 and eafib3 leading to ehr, ehf and econs, respectively, if afib is present as a patient feature. Other context processes are defined as in Section 5.

**Hypertension Guideline.** *Patient profile.* According to the event structure in Fig. 5, patient features of interest for the hypertension guideline are: hyper, diabetes, below55 (is below 55 years old), and origin (has African origins).

**Drugs.** Drugs in the hypertension guideline are represented by the following entities: captopril and benazepril (class acei), irbesartan and olmesartan (class arb), verapamil and diltiazem (class ccb), chlorothiazide and indapamide (class td).

**Therapies.** We just apply the context processes encoding defined in Section 5.

**Combined Guidelines Model.** The models of the AFib and hypertension guidelines share some entities because of common patient features (`hyper` and `diabetes`) used in guards of both guidelines, and because of drugs that could be prescribed for both diseases (`verapamil` and `diltiazem`, in class `ccb`). Moreover, the two patient features `over75` and `below55` describe conflicting age classes. Reactions and context processes handling the shared entities are included only once in the combined model, while the two context processes related to age classes are replaced by the single process  $K_{age} = \emptyset.Void + \{over75\}.Void + \{below55\}.Void$ .

**ADR table.** All the adverse drug reactions listed in Table 2 are included in the combined model, as explained in Section 5.

**Implementation of the Analyser.** In order to analyse the case study, we exploited BioReSolve [2], a tool implementing the SOS semantics of RSs. BioReSolve has been developed in SWI-Prolog [21] by some of the authors of this paper, according to the specifications in [8,10]. In addition, we extended BioReSolve by implementing the support for RS with Guards proposed in this paper, by following the definition given in Section 4.

The analysis of the AFib and hypertension guidelines is conducted by a Python script developed on purpose, which uses BioReSolve to generate the LTS that describes all possible treatments (according to the guidelines) for all possible patients. Python-Prolog interaction is made possible by the `swiplserver` Python package. The LTS obtained from BioReSolve is then processed by exploiting the Python package `networkx` to identify attractors (i.e., loops) and determine which patient profiles could lead to the prescription of a combination of drugs at risk for adverse reactions. The patient profiles characterized in this way are then expressed as a minimal logic formula in disjunctive normal form, suitable for tabular representation. Minimization of the logic formula is performed by the `logicmin` Python package that implements the Quine-McCluskey method. The

Patient profiles possibly leading to a "major" adverse reaction										
	afib	has_fib	heart_rate	consensus_acei	over75	below55	diabetes	doac_int	hyper	origin
1	TRUE		TRUE	TRUE	FALSE			FALSE	TRUE	
2	TRUE		TRUE	TRUE	FALSE		TRUE	FALSE		
3	TRUE		TRUE	TRUE	TRUE	FALSE		FALSE		
4	TRUE	TRUE		TRUE				FALSE	TRUE	
5	TRUE	TRUE							TRUE	
6	TRUE	TRUE	TRUE							

Patient profiles possibly leading to a "moderate" adverse reaction										
	afib	has_fib	heart_rate	consensus_acei	over75	below55	diabetes	doac_int	hyper	origin
1	TRUE		TRUE	TRUE	FALSE			FALSE	TRUE	
2	TRUE		TRUE	TRUE	FALSE		TRUE	FALSE		
3	TRUE		TRUE	TRUE	TRUE	FALSE		FALSE		
4	TRUE	TRUE		TRUE				FALSE	TRUE	
5	TRUE	TRUE		TRUE				FALSE	TRUE	
6	TRUE	TRUE	TRUE							

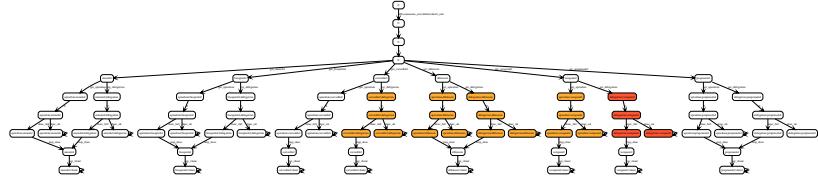
Patient profiles possibly leading to a "minor" adverse reaction										
	afib	has_fib	heart_rate	consensus_acei	over75	below55	diabete	doac_int	hyper	origin
1	TRUE		TRUE	TRUE	FALSE		FALSE		FALSE	

**Fig. 6.** Summary of patient profiles that could lead to treatments associated with a major, moderate and minor risk for adverse drug reactions, resp. Note that, by design, over75 and below55 cannot be co-present (TRUE) in the same profile.

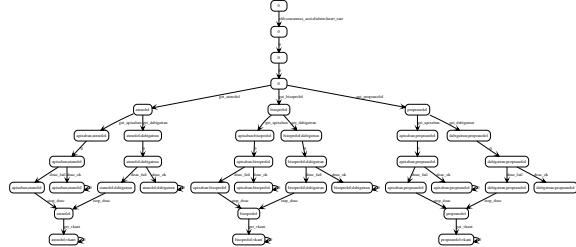
Python script we developed, along with the BioReSolve specification of the RS model to be analysed in this paper, are freely available at [14].

**Analysis of the Guidelines.** The analysis of the RS model of the two guidelines with BioReSolve and the Python script described above allows us to characterize the classes of patient profiles that could reach a state of major, moderate or minor risk of adverse reaction development. The LTS describing all treatment combinations for all patient profiles consists of 310k states and is constructed in 5 hours by our (non optimized) Python script running on a single core of a Intel i7 laptop. Logical formulas characterizing the three classes are represented in tabular form in Fig. 6. In the tables, columns correspond to patient features, and rows to patterns of patient profiles. A green cell (TRUE) in a row means that the corresponding feature is present, a red cell (FALSE) that the corresponding feature is absent, and a white cell that the presence of such a feature is irrelevant.

In order to show how the results obtained from our analysis method can contribute to optimize patient treatments and can suggest potential guidelines improvements, let us consider a couple of patient profiles identified as possibly leading to major adverse reactions. In particular, we consider the patient profile constituted by features afib, heart\_rate, consensus\_acei and diabetes. This patient profile matches the pattern in row 2 of the table for major in Fig. 6, as well as the one in row 2 of the table for moderate. In Fig. 7 we show the LTS for such a profile in which red and orange states confirm that both situations of moderate and major risks for adverse reactions can be reached. By inspecting the LTS it is possible to deduce that one of the ways to avoid these risk states is to prevent prescribing the carvedilol, diltiazem and verapamil drugs, in favour of atenolol, bisoprolol or propranolol. Indeed, if we slightly change the RS model by adding extra guards preventing the administration of the identified drugs, we obtain the LTS in Fig. 8, in which dangerous states can no longer be reached.



**Fig. 7.** LTS for patient profile: afib, heart\_rate, consensus\_acei, diabetes.



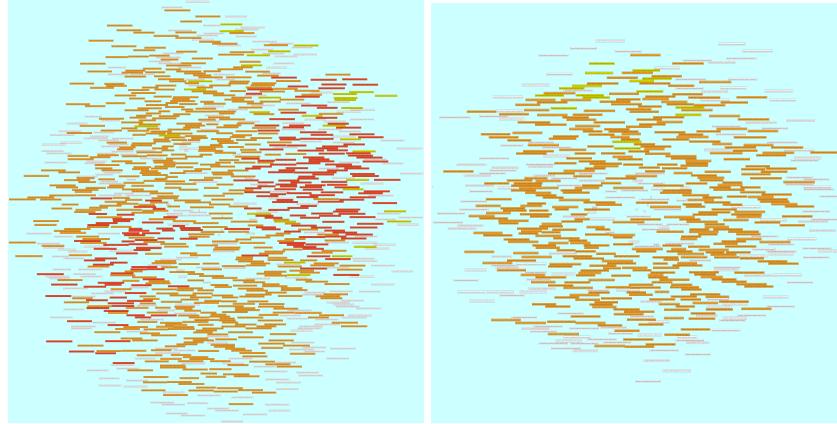
**Fig. 8.** LTS for the patient profile considered in Fig. 7, but in which the RS model has been modified to prevent prescribing carvedilol, diltiazem and verapamil.

This patient profile is a case in which only the AFib guideline is followed. Our analysis can be seen as a decision support approach for physicians, but also as a method to propose guideline refinements in a *precision medicine* style [1].

Let us now consider the patient profile constituted by the features `afib`, `heart_rate`, `consensus_acei` and `hyper`. This profile matches both the pattern in row 1 of the table for `major` in Fig. 6, and the one in row 1 of the table for `moderate`. This time the patient profile involves features from both guidelines, and this makes the corresponding LTS much more complex (more than 1.000 nodes and 2.300 arcs, see Fig. 9, left). The figure shows that several red states (associated to major risks for adverse reactions) can be reached. Nevertheless, a careful inspection of the LTS suggests that a cause of danger could be the administration of dabigatran. Again, a modification of the RS model preventing the prescription of such a drug in favour of apixaban (of the same class) allows states of major risk not to be reached, as shown in Fig. 9 (right), where red states are no longer present.

This second example of patient profile inspection shows that the proposed approach allows risks for adverse reactions to emerge before starting combinations of therapies from different guidelines for patients with comorbidities. This possibility opens to patient-targeted optimizations of combined treatments.

We remark that, although in this case study we resorted to visual inspection of the LTSs to identify causes of adverse reactions, the analysis could be automatized by applying, for instance, a slicing analysis method as we did in [13] for gene regulatory networks. Slicing analysis is currently able to find non-trivial causal relationships between reactions and produced entities in standard RSSs. For the analysis of guidelines, the method should be extended to deal with guards and with the ability to consider as causes for the production of an entity also choices done in context processes. This extension is left for future work.



**Fig. 9.** Left: LTS for patient profile: afib, heart\_rate, consensus\_acei, hyper. Right: same profile, but with RS model modified to avoid dabigatran.

## 8 Related Work and Concluding Remarks

Earlier work (see, e.g., [18,3,4,5,6]) explored the combination of formal verification techniques, such as constraint solvers and theorem provers, to identify steps in different guidelines that cause problems if carried out together and to determine optimal treatment paths with respect to a particular parameter (e.g., drug efficacy, cost, etc). However, none of them provided any visual aid to foresee and link current choices to future problems. They also lacked the ability to backtrack, to carry a more comprehensive analysis of treatment options (their advantages vs drawbacks) or to study different patient profiles. Indeed, earlier work focused entirely on the guidelines as given by the National Institute for Health and Care Excellence (NICE, see [nice.org.uk](http://nice.org.uk)) and the search through these in a more generic/idealised sense.

By contrast, here we model clinical guidelines directly with extended RSs, where contexts are guarded (recursive, nondeterministic) processes. Guarded contexts allow us to model (the administration of) different therapies, depending on the patients' profile. Despite our manual encoding of guidelines into RS processes, it is worth noting that their derivation can be fully automated. By running the RS specification in our toolkit (implemented by a prototype available in [14]), we derive a LTS for each patient profile representing all possible evolutions of patient therapies in accordance to the clinical guidelines. In this way, we can discover the choices in the (combined) treatments that can lead to critical situations and how to avoid them (preferably early on). To the best of our knowledge, this is the first approach with RSs to model clinical guidelines and patient therapies, including the effects of the combination of different therapies.

As future work, we envisage an interactive automated toolchain for generating RSs models from concise guideline descriptions, running the analysis of critical clinical paths and determining the main causes of dangerous side effects for different patient profiles, possibly reusing monitor-driven dynamic slicing techniques that are already available for RSs [10,11].

## References

1. Ashley, E.A.: Towards precision medicine. *Nature Reviews Genetics* **17**(9), 507–522 (2016). <https://doi.org/doi.org/10.1038/nrg.2016.86>
2. BioReSolve web page, a prolog interpreter for Reaction Systems analysis, <http://www.di.unipi.it/~bruni/LTSRS/>, accessed: May 3, 2024
3. Bowles, J., Caminati, M.: A flexible approach for finding optimal paths with minimal conflicts. In: Proceedings of ICFEM 2017. LNCS, vol. 10610, pp. 209–225. Springer (2017). [https://doi.org/10.1007/978-3-319-68690-5\\_13](https://doi.org/10.1007/978-3-319-68690-5_13)
4. Bowles, J., Caminati, M.: Balancing prescriptions with constraint solvers. In: Liò, P., Zuliani, P. (eds.) Automated Reasoning for Systems Biology and Medicine, Computational Biology, vol. 30, pp. 243–267. Springer (2019). [https://doi.org/10.1007/978-3-030-17297-8\\_9](https://doi.org/10.1007/978-3-030-17297-8_9)
5. Bowles, J., Caminati, M.: Correct composition in the presence of behavioural conflicts and dephasing. *Sci. Comput. Program.* **185** (2020). <https://doi.org/10.1016/j.scico.2019.102323>
6. Bowles, J., Caminati, M., Cha, S., Mendoza, J.: A framework for automated conflict detection and resolution in medical guidelines. *Sci. Comput. Program.* **182**, 42–63 (2019). <https://doi.org/10.1016/j.scico.2019.07.002>
7. Brijder, R., Ehrenfeucht, A., Main, M., Rozenberg, G.: A tour of reaction systems. *Int. J. Found. Comput. Sci.* **22**(07), 1499–1517 (2011). <https://doi.org/10.1142/S0129054111008842>
8. Brodo, L., Bruni, R., Falaschi, M.: A process algebraic approach to reaction systems. *Theoretical Computer Science* **881**, 62–82 (2021). <https://doi.org/10.1016/j.tcs.2020.09.001>
9. Brodo, L., Bruni, R., Falaschi, M.: A logical and graphical framework for reaction systems. *Theoretical Computer Science* **875**, 1–27 (2021). <https://doi.org/10.1016/j.tcs.2021.03.024>
10. Brodo, L., Bruni, R., Falaschi, M.: Dynamic slicing of reaction systems based on assertions and monitors. In: Proceedings of PADL 2023. LNCS, vol. 13880, pp. 107–124. Springer (2023). [https://doi.org/10.1007/978-3-031-24841-2\\_8](https://doi.org/10.1007/978-3-031-24841-2_8)
11. Brodo, L., Bruni, R., Falaschi, M.: A framework for monitored dynamic slicing of Reaction Systems. *Nat. Comput.* (2024). <https://doi.org/10.1007/s11047-024-09976-3>
12. Brodo, L., Bruni, R., Falaschi, M., Gori, R., Levi, F., Milazzo, P.: Quantitative extensions of reaction systems based on SOS semantics. *Neural Comput. Appl.* **35**(9), 6335–6359 (2023). <https://doi.org/10.1007/s00521-022-07935-6>
13. Brodo, L., Bruni, R., Falaschi, M., Gori, R., Milazzo, P.: Attractor and slicing analysis of a T Cell differentiation model based on reaction systems. In: Proceedings of DataMod 2023. LNCS, Springer Berlin (in press)
14. Github repository with the RS model and Python script developed for this paper, <https://github.com/Unipisa/AFib-Hyper-GuidelinesAnalysis>, accessed: May 3, 2024
15. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. *Fundam. Inf.* **75**(1–4), 263–280 (2007), <http://content.iospress.com/articles/fundamenta-informaticae/fi75-1-4-15>
16. Feder, G., Eccles, M., Grol, R., Griffiths, C., Grimshaw, J.: Using clinical guidelines. *Bmj* **318**(7185), 728–730 (1999). <https://doi.org/10.1136/bmj.318.7185.728>
17. Hughes, L., McMurdo, M.E.T., Guthrie, B.: Guidelines for people not for diseases: the challenges of applying UK clinical guidelines to people with multimorbidity. *Age and Ageing* **42**, 62–69 (2013). <https://doi.org/10.1093/ageing/afs100>

18. Kovalov, A., Bowles, J.: Avoiding medication conflicts for patients with multimorbidities. In: Proceedings of iFM 2016. LNCS, vol. 9681, pp. 376–392. Springer (2016). [https://doi.org/10.1007/978-3-319-33693-0\\_24](https://doi.org/10.1007/978-3-319-33693-0_24)
19. Milner, R.: A Calculus of Communicating Systems, LNCS, vol. 92. Springer (1980)
20. Plotkin, G.D.: A structural approach to operational semantics. *J. Log. Algebraic Methods Program.* **60-61**, 17–139 (2004)
21. SWI-Prolog home page, <https://www.swi-prolog.org/>, accessed: 3 May 2024
22. Woolf, S.H., Grol, R., Hutchinson, A., Eccles, M., Grimshaw, J.: Potential benefits, limitations, and harms of clinical guidelines. *Bmj* **318**(7182), 527–530 (1999). <https://doi.org/10.1136/bmj.318.7182.527>