Deciding the Precongruence for Deadlock Freedom Using Operating Guidelines

Richard Müller[1,2] and Christian Stahl[2]

[1] Institut für Informatik, Humboldt-Universität zu Berlin, Germany
richard.mueller@informatik.hu-berlin.de
[2] Department of Mathematics and Computer Science,
Technische Universiteit Eindhoven, The Netherlands
c.stahl@tue.nl

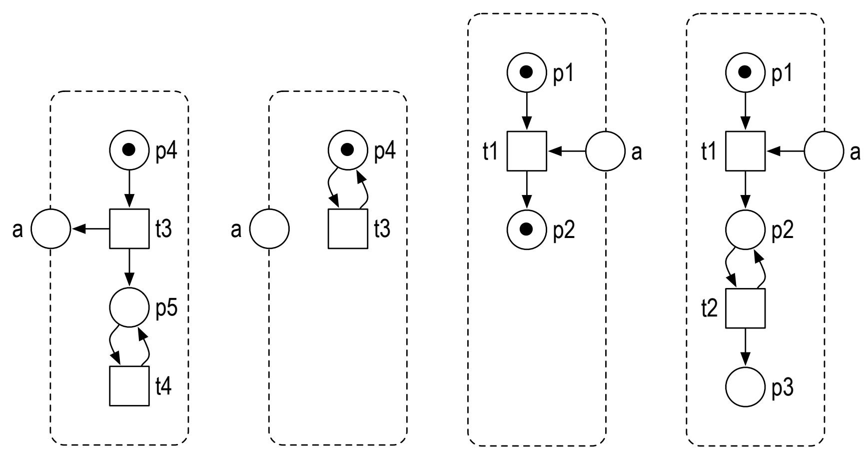
Abstract. In the context of asynchronously communicating and deadlock free services, the refinement relation of services has been formalized by the *accordance preorder*. A service *Impl* accords with a service *Spec* if every *controller* of *Spec*—that is, every environment that can interact with service *Spec* without deadlocking—is a controller of *Impl*. The procedure to decide accordance of two services uses that the set of controllers of a finite-state service has a finite representation, called *operating guideline*. Recently, it has been shown that the accordance preorder is not a precongruence and thus the decision procedure based on operating guidelines cannot be used. In this paper, we *adapt the results on operating guidelines to the precongruence setting*: We define an operating guideline that represents all controllers of a service w.r.t. the accordance precongruence and show how this refinement relation of two services can be decided based on their operating guidelines.

1 Introduction

Service-oriented computing (SOC) [6] aims at building complex systems by aggregating less complex, independently-developed building blocks called *services*. A service is an autonomous system that has an interface to interact with other services via asynchronous message passing. Designing a system in such a way allows for rapidly adjusting it to prevalent needs. Services sometimes need to be replaced—for example, when new features have been implemented or bugs have been fixed. This requires a notion of service *refinement*, which should, according to the idea of SOC, respect *compositionality*: If a service *Impl* refines a service *Spec*, then any environment that can correctly interact with *Spec* can also correctly interact with *Impl*. We refer to such an environment as a *controller* of *Impl* and *Spec*, respectively. Compositionality is crucial, because organizations usually do not know the services of other organizations involved in the system.

The absence of deadlocks is a commonly agreed minimal requirement for the behavioral correctness of a service-oriented system. Stahl et al. [7] formalized the replacement (or refinement) relation in the context of deadlock freedom by the *accordance* preorder. The decision procedure uses that, for finite-state services with bounded buffers, the set of controllers has a finite representation, the *operating guideline* [4] of the service. The decision procedure in [7] has two inherent characteristics: First, the interior of a service must be bounded when considered in isolation. Second, it allows for two possibly different bounds: one for the buffers and one for the interior of a service.

Recently, Stahl and Vogler [8] introduced a modified accordance relation which differs from the original accordance relation in two ways: First, the modified accordance relation has been proven to be a *precongruence* w.r.t. service composition; that is, it respects compositionality. Second, the modified accordance relation is more uniform than the original accordance relation in [7]: Stahl and Vogler [8] do not require the interior of a service to be bounded when considered in isolation and prescribe only one bound for the buffers and for the interior of a service rather than possible different bounds as in [7].



(a) Open net *Spec* **(b)** Open net *Impl* **(c)** Open net N_1 **(d)** Open net N_2

Fig. 1. Open net *Impl* accords with open net *Spec* but not vice versa.

We illustrate the difference between the accordance relation in [7] and the precongruence in [8] with an example: Figure 1 depicts four services modeled as open nets. As shown in [8], open net *Impl* accords with open net *Spec* for a bound $b = 1$ if we consider the precongruence, but *Spec* does not accord with *Impl*. To see this, consider the open net N_1 in Fig. 1(c) and compose N_1 with *Spec* and *Impl* by merging the common interface places a. The composition of *Impl* and N_1 has only one reachable marking, $[p_1, p_2, p_4]$, in which transition t_3 is continuously enabled. Thus, the composition is deadlock free and N_1 is a controller of *Impl*. Now consider the composition of *Spec* and N_1. It has a reachable marking where p_2 contains two tokens. Thus, the composition is not 1-bounded and N_1 is not a controller (for a bound of 1) of *Spec*. Similarly, open net N_2 in Fig. 1(d) is a controller of *Impl* but not a controller of *Spec* (for a bound of 1), because p_3 is unbounded in the composition of *Spec* and N_2.

However, applying the decision procedure in [7] based on operating guidelines, *Spec* and *Impl* are even accordance equivalent (assuming a single bound for the interface and the interior); that is, every controller of *Impl*—like the open net N_1 or N_2—is also a controller of *Spec*. The cause for this result is that [7] does not consider N_1 and N_2, because their interiors are not 1-bounded.

So the example shows, if we assume a single bound for the interface and the interior of a service, then the accordance precongruence implies accordance but not the other way around. The reason is that the precongruence is more uniform and considers a more general notion of a service. If we consider different bounds for the interface and the interior of a service, then both refinement relations are incomparable.

Stahl and Vogler [8] presented a procedure to decide the accordance precongruence, but they also showed that the accordance precongruence cannot be decided using the procedure in [7] based on operating guidelines without adaptation. In this paper, we present an operating guideline representing the set of all controllers in the precongruence setting of [8] and show how this operating guideline can be used to decide accordance of two services. Our motivation for adapting the theory of operating guidelines from the setting of [7] to the setting of [8] is twofold: First, we want to present the theory for deciding accordance using operating guidelines such that the existing implementation in the tool Cosme [5] can be reused and that the technique can also be applied in the precongruence setting. Second, operating guidelines have proved their usefulness also in other applications than deciding accordance, including service correction [3], test case generation [1], and instance migration [2]. As the more general notion of a controller is advantageous also for those applications, extending the theory on operating guidelines is natural.

This paper is organized as follows: Section 2 introduces open nets, our formal model for services, and gives some background information. Section 3 introduces operating guidelines and adapts the matching technique to the modified accordance relation. Section 4 decides the precongruence for deadlock freedom using operating guidelines. We close with a discussion of related work and a conclusion in Sect. 5.

2 Preliminaries

This section provides the basic notions, such as Petri nets, open nets for modeling services, and open net environments for describing the behavior of open nets.

For two sets A and B, let $A \uplus B$ denote the disjoint union; writing $A \uplus B$ expresses the implicit assumption that A and B are disjoint. Let $\mathbb{N}$ denote the non-negative integers, and let $\mathbb{N}^+$ denote the positive integers. For a set A, let $\mathcal{P}(A)$ denote the powerset of A, and let $|A|$ denote the cardinality of A.

2.1 Petri Nets

As a basic model, we use place/transition Petri nets extended with a set of final markings and transition labels.

Definition 1 (net). A *net* $N = (P, T, F, m_N, \Omega)$ consists of

- a finite set P of *places*,
- a finite set T of *transitions* such that P and T are disjoint,
- a *flow relation* $F \subseteq (P \times T) \uplus (T \times P)$,
- an *initial marking* m_N, where a marking is a mapping $m : P \to \mathbb{N}$, and
- a set Ω of *final markings*.

A *labeled net* $N = (P, T, F, m_N, \Omega, \Sigma_{in}, \Sigma_{out}, l)$ is a net (P, T, F, m_N, Ω) together with an *alphabet* $\Sigma = \Sigma_{in} \uplus \Sigma_{out}$ of *input actions* Σ_{in} and *output actions* Σ_{out} and a *labeling function* $l : T \to \Sigma \uplus \{\tau\}$, where τ represents an invisible, internal action.

In this paper, we only treat labeled nets where, for every transition t, the *label* $l(t)$ of t is either τ or t itself.

Introducing net N implicitly introduces its components P, T, F, m_N, Ω; the same applies to nets N', N_1, etc. and their components $P', T', F', m_{N'}, \Omega'$, and $P_1, T_1, F_1, m_{N_1}, \Omega_1$, respectively—and it also applies to other structures later on.

Graphically, a circle represents a place, a box represents a transition, and the directed arcs between places and transitions represent the flow relation. A marking is a distribution of tokens over the places. Graphically, a black dot represents a token. Transition labels beside τ are written into the respective boxes.

Let $x \in P \uplus T$ be a node of a net N. As usual, ${}^\bullet x = \{y \mid (y, x) \in F\}$ denotes the *preset* of x and $x^\bullet = \{y \mid (x, y) \in F\}$ the *postset* of x. We canonically extend the notion of a preset/postset to sets of nodes. We interpret presets and postsets as multisets when used in operations also involving multisets. A marking is a multiset over the set P of places; for example, $[p_1, 2p_2]$ denotes a marking m with $m(p_1) = 1$, $m(p_2) = 2$, and $m(p) = 0$ for $p \in P \setminus \{p_1, p_2\}$. For $n \in \mathbb{N}$, a place $p \in P$ and a set M of markings over P, $M(p) = n$ denotes that for all $m \in M$, $m(p) = n$. We define $+$ and $-$ for the sum and the difference of two markings and $=, <, >, \leq, \geq$ for comparison of markings in the standard way. We canonically extend the notion of a marking of N to supersets $Q \supseteq P$ of places; that is, for a mapping $m : P \to \mathbb{N}$, we extend m to the marking $m : Q \to \mathbb{N}$ such that for all $p \in Q \setminus P$, $m(p) = 0$. Analogously, a marking can be restricted to a subset $Q \subseteq P$ of the places of N.

The *behavior* of a net N relies on the marking of N and changing the marking by the firing of transitions of N. A transition $t \in T$ is *enabled* at a marking m, denoted by $m \xrightarrow{t}$, if for all $p \in {}^\bullet t$, $m(p) > 0$. If t is enabled at m, it can *fire*, thereby changing the marking m to a marking $m' = m - {}^\bullet t + t^\bullet$. The firing of t is denoted by $m \xrightarrow{t} m'$; that is, t is enabled at m and firing it results in m'. The behavior of N can be extended to sequences: $m_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{k-1}} m_k$ is a *run* of

N if for all $0 < i < k$, $m_i \xrightarrow{t_i} m_{i+1}$. A marking m' is *reachable from* a marking m if there exists a (possibly empty) run $m_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{k-1}} m_k$ with $m = m_1$ and $m' = m_k$; for $v = t_1 \ldots t_k$, we also write $m_1 \xrightarrow{v} m_k$. Marking m' is *reachable* if $m_N = m$. The set M_N represents the set of all reachable markings of N.

In the case of labeled nets, we lift runs to traces: If $m_1 \xrightarrow{v} m_k$ and w is obtained from v by replacing each transition by its label and removing all τ labels, we write $m_1 \xRightarrow{w} m_k$ and refer to w as a *trace*. As usual, ε denotes the empty trace. The *reachability graph* $RG(N)$ of net N has the reachable markings M_N as its nodes and a t-labeled edge from m to m' whenever $m \xrightarrow{t} m'$ in N. In the case of a labeled net, each edge label t is replaced by $l(t)$.

Finally, we introduce b-boundedness and deadlock freedom of nets. A marking m of net N is *b-bounded* for a bound $b \in \mathbb{N}^+$, if $m(p) \leq b$ for all $p \in P$. Net N is *b-bounded* if every reachable marking is b-bounded. The set M_N^b represents the set of all reachable b-bounded markings of N. A reachable marking $m \notin \Omega$ of N is a *deadlock* if no transition $t \in T$ of N is enabled at m. If N has no deadlock, then it is deadlock free.

2.2 Open Nets and Open Net Behavior

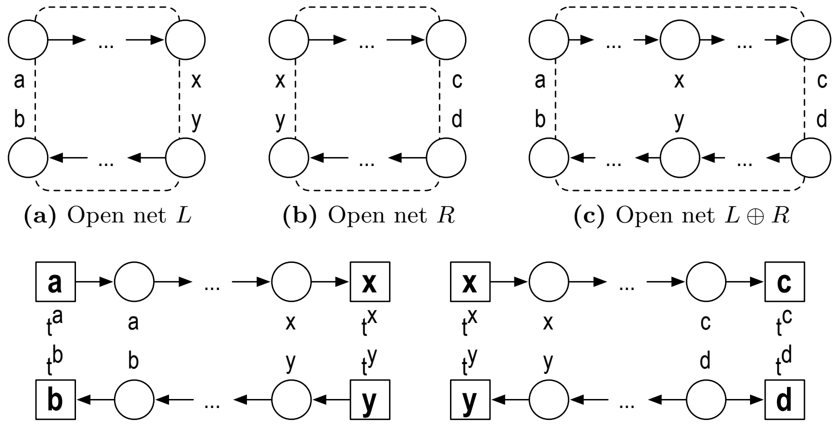
Like Lohmann et al. [4] and Stahl et al. [7], we model services as *open nets* [9,4], thereby restricting ourselves to the communication protocol of a service. In the model, we abstract from data and identify each message by the label of its message channel. An open net extends a net by an interface. An interface consists of two disjoint sets of input and output places corresponding to asynchronous input and output channels. In the initial marking and the final markings, interface places are not marked. An input place has an empty preset, and an output place has an empty postset.

Definition 2 (open net). An *open net* N is a tuple $(P, T, F, m_N, \Omega, I, O)$ with

- $(P \uplus I \uplus O, T, F, m_N, \Omega)$ is a net,
- for all $p \in I \uplus O$, $m_N(p) = 0$ and $\Omega(p) = 0$,
- the set I of *input places* satisfies ${}^\bullet I = \emptyset$, and
- the set O of *output places* satisfies $O^\bullet = \emptyset$.

If $I = O = \emptyset$, then N is a *closed net*. Open net N is *sequentially communicating* if each transition is connected to at most one *interface place* $I \uplus O$. The *inner* net *inner*(N) results from removing the interface places and their adjacent arcs from N. Two open nets are *interface equivalent* if they have the same sets of input and output places.

Graphically, we represent an open net like a net with a dashed frame around it. The interface places are depicted on the frame. Later, we consider the behavior of an open net, which is basically its reachability graph. To simplify the labeling of transitions connected to interface places, we only consider sequentially communicating nets. That way, each transition is labeled by a single label



(a) Open net L **(b)** Open net R **(c)** Open net $L \oplus R$

(d) Labeled net $env(L)$ **(e)** Labeled net $env(R)$

Fig. 2. Schematic example of open nets, open net composition, and their environment.

rather by a set of labels. This restriction is not significant as every open net can be transformed into an equivalent sequentially communicating open net [4].

For the composition of open nets, we assume that the sets of transitions are pairwise disjoint and that no internal place of an open net is a place of any other open net. In contrast, the interfaces intentionally overlap. We require that all communication is *bilateral* and *directed*; that is, every shared place p has only one open net that sends into p and one open net that receives from p. We refer to open nets that fulfill these properties as *composable*. We compose two composable open nets N_1 and N_2 by merging shared interface places and turn these places into internal places; see Fig. 2(a) and 2(b) for a schematic example of open nets and their composition. The definition of composable thereby guarantees that an open net composition is again an open net (possibly a closed net).

Definition 3 (open net composition). Open nets N_1 and N_2 are *composable* if $(P_1 \uplus T_1 \uplus I_1 \uplus O_1) \cap (P_2 \uplus T_2 \uplus I_2 \uplus O_2) = (I_1 \cap O_2) \uplus (I_2 \cap O_1)$. The *composition* of two composable open nets N_1 and N_2 is the open net $N_1 \oplus N_2 = (P, T, F, m_N, \Omega, I, O)$ where

- $P = P_1 \uplus P_2 \uplus (I_1 \cap O_2) \uplus (I_2 \cap O_1)$,
- $T = T_1 \uplus T_2$,
- $F = F_1 \uplus F_2$,
- $m_N = m_{N_1} + m_{N_2}$,
- $I = (I_1 \uplus I_2) \setminus (O_1 \uplus O_2)$,
- $O = (O_1 \uplus O_2) \setminus (I_1 \uplus I_2)$, and
- $\Omega = \{m_1 + m_2 \mid m_1 \in \Omega_1, m_2 \in \Omega_2\}$.

To define the *behavior* of an open net N, we consider its environment $env(N)$. The net $env(N)$ is a net that can be constructed from N by adding to each interface place $p \in I \uplus O$ a p-labeled transition t^p in $env(N)$. The net $env(N)$ is just a tool to define our characterizations and prove our results. Intuitively, one can understand the construction as translating the asynchronous interface of N into a buffered synchronous interface (with unbounded buffers) described by the transition labels of $env(N)$.

Definition 4 (open net environment). The *environment* of an open net N is the labeled net $env(N) = (P \uplus I \uplus O, T \uplus T', F \uplus F', m_N, \Omega, I, O, l)$ where

- $T' = \{t^x \mid x \in I \uplus O\}$ is the set of *interface transitions*,
- $F' = \{(t^x, x) \mid x \in I\} \uplus \{(x, t^x) \mid x \in O\}$, and
- $l(t) = \begin{cases} \tau, & t \in T \\ x, & t^x \in T'. \end{cases}$

We refer to a transition from T as *internal transition*. A marking m of $env(N)$ is *stable* if at most internal transitions of $env(N)$ are enabled at m.

Figures 2(d) and 2(e) show the environments of the open nets L and R from Fig. 2(a) and 2(b). A transition label is depicted inside a transition with bold font to distinguish it from the transition's identity.

The behavior of an open net N can now be defined by the reachability graph $RG(env(N))$ of its environment. As we are interested in finite-state services, we always define the behavior of an open net with regard to a bound b. As soon as b is violated, we can stop the computation of the behavior in this state; however, we keep this state to identify the bound violation.

Definition 5 (open net behavior). Let $b \in \mathbb{N}^+$. The *b-behavior* $beh_b(N)$ of an open net N is the reachability graph of $env(N)$ where we remove all outgoing edges from every non-b-bounded node (thereby removing unreachable nodes and edges too).

Clearly, the b-behavior of an open net N has at most $(b+2)^{(|P|+|I|+|O|)}$ states.

Figure 3 depicts the environment net of open net N_2 and its behavior $beh_1(N_2)$. Recall that transitions t_1 and t_2 are labeled τ. Every leaf in $beh_1(N_2)$ violates the bound and has thus no successor.

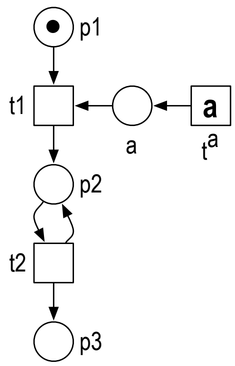
We interpret $beh_b(N)$ as a labeled automaton with input and output labels.

Definition 6 (automaton). An *automaton* $A = (Q, E, q_A, \Sigma_{in}, \Sigma_{out})$ consists of

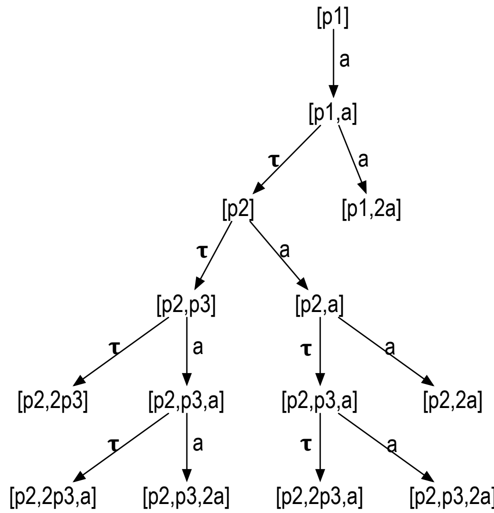
- a finite set Q of *states*,
- an *edge relation* $E \subseteq Q \times \big(\Sigma_{in} \uplus \Sigma_{out} \uplus \{\tau\}\big) \times Q$,
- an *initial node* q_A, and
- an *alphabet* $\Sigma = \Sigma_{in} \uplus \Sigma_{out}$ of *input labels* Σ_{in} and *output labels* Σ_{out}.

A is *deterministic* if no node has two outgoing edges with the same label.

We compare two automata with a simulation relation, thereby treating τ as an ordinary action.



(a) Labeled net $env(N_2)$



(b) Behavior $beh_1(N_2)$

Fig. 3. Constructing the 1-behavior of open net N_2.

Definition 7 (simulation relation). Let A and B be two automata with label set $\Sigma = \Sigma_{in} \uplus \Sigma_{out}$. Then $\varrho \subseteq Q_A \times Q_B$ is a *simulation* of A by B if

– $(q_A, q_B) \in \varrho$, and
– for every $(p, q) \in \varrho$, $x \in \Sigma \uplus \{\tau\}$, $p' \in Q_A$ such that $p \xrightarrow{x} p'$ in A, there exists $q' \in Q_B$ such that $q \xrightarrow{x} q'$ in B and $(p', q') \in \varrho$.

Simulation ϱ is *minimal* if for every simulation ϱ' of A by B, $\varrho \subseteq \varrho'$.

For all automata A and B where B is deterministic, the minimal simulation relation of A by B is uniquely defined.

3 Operating Guidelines

In this section, we formally define the notion of a controller of an open net N and present a finite representation of all controllers of N, the *operating guideline* of N.

The composition of a service C with a service N shall be deadlock free; that is, if the composition gets stuck, then it is in a final state. As we are interested in finite-state services, the composition must be bounded. A service C guaranteeing these two requirements can be seen as a *controller* of the service N.

Definition 8 (b-controller). Let $b \in \mathbb{N}^+$. An open net C is a *b-controller* of an open net N if the composition $N \oplus C$ is a closed net, deadlock free, and b-bounded.

A b-operating guideline $OG_b(N)$ of a service N describes how another service C should successfully communicate with N. Technically, it characterizes the possibly infinite set of b-controllers of N in a finite manner. Because a b-controller of N provides suitable inputs for N and accepts its outputs, $OG_b(N)$ interchanges the inputs and outputs of N. The structure of $OG_b(N)$ is an automaton where a Boolean formula is attached to each state. The structure is the behavior of a b-controller that exhibits the behavior of every b-controller of N; the formula of a state indicates which combinations of outgoing edges must be present in any b-controller. Thus, a literal of such a Boolean formula is a transition label of N or the literal *final*, specifying that N is in a final state. That way, we can employ simulation for comparing the behavior of an open net with $OG_b(N)$ later on.

Definition 9 (annotated automaton). An *annotated automaton* $(Q, E, q_A, \Sigma_{in}, \Sigma_{out}, \phi)$ is an automaton $(Q, E, q_A, \Sigma_{in}, \Sigma_{out})$ whose nodes $q \in Q$ are annotated with a *Boolean formula* $\phi(q)$ over $\Sigma_{in} \uplus \Sigma_{out} \uplus \{\mathit{final}\}$.

To construct $OG_b(N)$, we calculate the b-behavior $\mathit{beh}_b(N)$ of N and make the automaton deterministic by constructing the powerset automaton. A state of $OG_b(N)$ contains a set of markings of $\mathit{env}(N)$; we refer to it as a *node*. These markings can be reached by firing internal transitions of $\mathit{env}(N)$. An edge connects two nodes of $OG_b(N)$, thereby referring to an interface transition of $\mathit{env}(N)$ (i.e., the environment takes a token from an output place or produces a token on an input place of N). A b-controller cannot know which marking m of a node Q net $\mathit{env}(N)$ might be in, but it has to avoid a deadlock and a bound violation in any case; the formula $\phi(Q)$ describes how to do this. The literals of ϕ are $I \uplus O \uplus \{\mathit{final}\}$. Recall that nonstable markings have an internal transition enabled and, thus, are not deadlocks; all internal transitions remain in the same node. As a consequence, $\phi(Q)$ is a *conjunction indexed by all stable markings* $m \in Q$. Every conjunct is a disjunction of the following propositional atoms: *final* if m is a final marking, $x \in I$ if $Q \xrightarrow{x}$ (i.e., x does not lead to a bound violation in any case), and $x \in O$ if t^x is enabled at m (i.e., if in marking m, net N has already produced a message on output place x). Hence, the formulae are in conjunctive normal form (CNF) without negation. Here, $Q \xrightarrow{x}$ means that Q has an outgoing x-labeled edge.

Definition 10 (b-operating guideline). Let $b \in \mathbb{N}^+$. The *b-operating guideline* of an open net N is the annotated automaton $OG_b(N) = (\mathcal{Q}, E, Q_0, \Sigma_{in}, \Sigma_{out}, \phi)$, where

- $\mathcal{Q} = \mathcal{P}(M^b_{\mathit{env}(N)})$ is a set of *nodes*,
- $E = \{(Q, x, Q') \in \mathcal{Q} \times I \uplus O \times \mathcal{Q} \mid Q' = \{m' \mid \exists m \in Q : m \stackrel{x}{\Longrightarrow} m'\}\} \uplus \{(Q, \tau, Q) \mid Q \in \mathcal{Q}\}$ is a set of *edges*,
- $Q_0 = \{m' \mid m_{\mathit{env}(N)} \stackrel{\varepsilon}{\Longrightarrow} m'\} \cap \mathcal{P}(M^b_{\mathit{env}(N)})$ is the *initial node*,
- $\Sigma_{in} = O$ are the *input labels*,
- $\Sigma_{out} = I$ are the *output labels*, and

– ϕ associates to each $Q \in \mathcal{Q}$ a *Boolean formula* with propositional atoms taken from $I \uplus O \uplus \{\mathit{final}\}$ such that

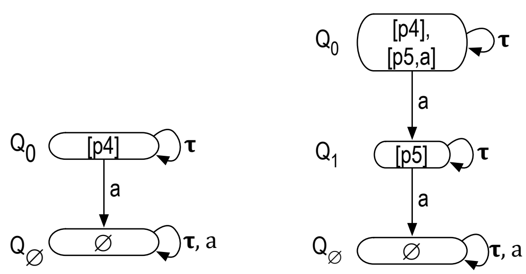
$$\phi(Q) = \bigwedge_{m : m \in Q \wedge m \text{ is stable}} \big(\psi_1(m) \vee \psi_2(m)\big) \quad \text{with}$$

$$\psi_1(m) = \bigvee_{x : x \in I \wedge Q \xrightarrow{x}} x \;\vee \bigvee_{x : x \in O \wedge m \xrightarrow{t^x}} x$$

$$\psi_2(m) = \begin{cases} \mathit{final}, & \text{if } m \in \Omega_{env(N)}, \\ \mathit{false}, & \text{otherwise.} \end{cases}$$

Clearly, $OG_b(N)$ is finite and deterministic by construction; if $Q_0 = \emptyset$, then the b-operating guideline of N does not exist. We refer to $Q \in \mathcal{Q}$ with $Q = \emptyset$ as the *empty node* and denote it by $Q_\emptyset$. Intuitively, the empty node $Q_\emptyset$ refers to markings which are unreachable in $env(N)$.

We proceed with a short complexity analysis. Let $b \in \mathbb{N}^+$ and N be an open net. Let further $x = |M^b_{env(N)}|$ denote the cardinality of the set of reachable, b-bounded markings of $env(N)$, and let $k = |I \uplus O|$ denote the size of the interface. The powerset construction may yield, in worst case, 2^x nodes of $OG_b(N)$. The formula $\phi(Q)$ of a node Q has at most $x \cdot (k+1)$ literals. As calculating the formula of a node can be done during the construction, $OG_b(N)$ can be computed in time and space proportional to $O(2^x \cdot x \cdot (k+1))$.



(a) $OG_1(\mathit{Impl})$ **(b)** $OG_1(\mathit{Spec})$

Fig. 4. Operating guidelines of open nets *Impl* and *Spec*. The annotation of all nodes is *true*, which we omitted.

Figure 4 depicts the 1-operating guidelines for open nets *Spec* and *Impl*. All nodes of $OG_1(\mathit{Impl})$ and $OG_1(\mathit{Spec})$ have the same annotation, *true*[3], thus we omitted them. For $OG_1(\mathit{Impl})$, we have $Q_0 = \{[p_4]\}$. A 1-controller can receive

[3] An annotation is a formula over $I \uplus O \uplus \{\mathit{final}\}$; *true* and *false* are also Boolean formulae.

message a, but *Impl* will never send this message. Thus, there is an a-labeled edge from Q_0 to the empty node $Q_\emptyset$. In $Q_\emptyset$, every action can occur, because the empty node refers to markings which are unreachable in $env(Impl)$.

We determine if an open net C is a b-controller of an open net N by *matching* its b-behavior $beh_b(C)$ with the b-operating guideline $OG_b(N)$ of N. To this end, we need to check whether C and N are composable, the behavior of C can be mimicked by $OG_b(N)$ (by checking a simulation relation), and every state m of $beh_b(C)$ satisfies the Boolean formula in the corresponding node Q of $OG_b(N)$. State m satisfies $\phi(Q)$ if either a correct combination of interface transition of $env(C)$ is enabled at m such that $N \oplus C$ remains b-bounded or m is a final marking and $env(N)$ is in a final marking, too (i.e., $\phi(Q)$ contains the literal *final*).

Definition 11 (matching). Let $b \in \mathbb{N}^+$ and let N and C be composable open nets. Then $beh_b(C)$ *matches* with $OG_b(N)$ if

1. The input (output) labels of $beh_b(C)$ are the input (output) labels of $OG_b(N)$.
2. There exists a minimal simulation relation ϱ of $beh_b(C)$ by $OG_b(N)$ such that
   (a) if $[m, Q] \in \varrho$ with m not b-bounded in $env(C)$, then $Q = Q_\emptyset$, and
   (b) if $[m, Q] \in \varrho$ with m stable in $env(C)$, then $\phi(Q)$ evaluates to *true*, written $m \models \phi(Q)$, for the following assignment β:
      – $\beta(c) = \mathit{true}$ if $c \neq \mathit{final}$ and $m \xrightarrow{c}$ in $beh_b(C)$,
      – $\beta(c) = \mathit{true}$ if $c = \mathit{final}$ and $m \in \Omega_{env(C)}$, and
      – $\beta(c) = \mathit{false}$, otherwise.

Consider again open net N_2, which is a 1-controller of *Impl*. Automaton $beh_1(N_2)$ in Fig. 3(b)) matches with $OG_1(Impl)$ (Fig. 4(a)). The simulation relation relates state $[p1]$ with Q_0 and all other states of $beh_1(N_2)$ with $Q_\emptyset$. The annotations trivially evaluate to *true*. Open net N_2 is not a 1-controller of *Spec* and $beh_1(N_2)$ does not match with $OG_1(Spec)$: The simulation relation relates state $[p_2, 2p_3]$ with node Q_1, thereby violating item 2(a) of Def. 11.

With the next theorem, we show that the b-operating guideline of an open net N characterizes the set of b-controllers of N.

Theorem 12 (b-controllability vs. matching). *Let $b \in \mathbb{N}^+$. For composable open nets N and C, C is a b-controller of N iff $beh_b(C)$ matches with $OG_b(N)$.*

Proof. ($\Rightarrow$): Let C be a b-controller of N. Then item (1) of Def. 11 holds because C and N are composable and $N \oplus C$ is a closed net.

Suppose a simulation relation ϱ of $beh_b(C)$ by $OG_b(N)$ does not exist. Then there exists $(m, Q) \in \varrho$ and $m \xrightarrow{x}$ in $beh_b(C)$ but $Q \overset{x}{\not\longrightarrow}$ in $OG_b(N)$ by Def. 7. By Def. 10, $Q \xrightarrow{x} Q'$ and there exists a marking of $env(N)$ in Q' that violates bound b and, therefore, Q' has been removed from $OG_b(N)$. As the respective trace to Q' is also a trace in $beh_b(C)$, there is a corresponding marking in $M_{N \oplus C}$ that violates the bound, and we have a contradiction to our assumption. Thus, ϱ exists, and ϱ is even minimal as $OG_b(N)$ is deterministic by Def. 10.

To show item (2a) of Def. 11, assume $(m, Q) \in \varrho$, with m is not b-bounded, and $Q \neq Q_\emptyset$. There exists $v \in (I \uplus O)^*$ with $m_{env(C)} \stackrel{v}{\Longrightarrow} m$ in $env(C)$ by Def. 5 and $m_{env(N)} \stackrel{v}{\Longrightarrow} m'$ in $env(N)$ by Def. 10. As a consequence, we find a corresponding marking in $M_{N \oplus C}$ that is not b-bounded; thus, we have a contradiction to our assumption and conclude $Q = Q_\emptyset$.

To show item (2b) of Def. 11, let $(m, Q) \in \varrho$ such that m is stable in $env(C)$. We show for each $m' \in Q$ with m' is stable in $env(N)$ that $m \models \psi_1(m') \vee \psi_2(m')$. If $m + m' \in \Omega_{N \oplus C}$, then $m \in \Omega_{env(C)}$ and $\psi_2(m') = \mathit{final}$, thus $m \models \psi_2(m')$ by Def. 11. Assume $m + m' \notin \Omega_{N \oplus C}$. Then C can either produce a token on a place $i \in I_N$ or consume a token from a place $o \in O_N$, because $N \oplus C$ is deadlock free by assumption. In the former case, we have $m \xrightarrow{i}$ in $beh_b(C)$, and $Q \xrightarrow{i}$ as $N \oplus C$ is b-bounded. Thus, $m \models \psi_1(m')$ by Def. 11. In the latter case, we have $m \xrightarrow{o}$ in $beh_b(C)$, and $m' \xrightarrow{t^o}$. Thus, $m \models \psi_1(m')$ by Def. 11.

($\Leftarrow$): Let ϱ be a minimal simulation of $beh_b(C)$ by $OG_b(N)$. We have to show that $N \oplus C$ is a closed net, deadlock free, and b-bounded.

$N \oplus C$ is a closed net because of item (1) in Def. 11. Next, we show that $N \oplus C$ is b-bounded. Let m (m') be a marking of C (N) such that $m + m'$ is a reachable marking of $N \oplus C$ that violates the bound. Let v denote the trace of $env(C)$ that corresponds to the run from m_C to m. As ϱ exists, v is also a trace in $OG_b(N)$ and so it is in $env(N)$. By the construction of $OG_b(N)$, the corresponding markings in $env(N)$ do not violate the bound, so it suffices to assume that m violates the bound in $env(C)$. Then, $(m, Q) \in \varrho$ with $Q = Q_\emptyset$ by assumption. However, this implies that $m + m'$ is not reachable in $M_{N \oplus C}$, which is a contradiction to our assumption. Thus, $N \oplus C$ is b-bounded.

Finally, we show that $N \oplus C$ is deadlock free. Let m (m') be a marking of C (N) such that $m + m'$ is a reachable marking of $N \oplus C$. Marking m is also a state in $beh_b(C)$. From the existence of ϱ we conclude that there exists a node Q of $OG_b(N)$ with $(m, Q) \in \varrho$. Further, we have $m' \in Q$; otherwise, $N \oplus C$ is not b-bounded. Assume m is stable in $env(C)$ and m' is stable in $env(N)$; otherwise, $m + m'$ is no deadlock of $N \oplus C$ by Def. 4. Then $m \models \psi_1(m') \vee \psi_2(m')$ by assumption. If $m \models \psi_1(m')$, then there exists $x \in (I \uplus O)$ with $m \xrightarrow{x}$ in $beh_b(C)$ by Def. 11. The corresponding transition is also enabled in $N \oplus C$; thus, $m + m'$ is no deadlock. If $m \models \psi_2(m')$, then $m \in \Omega_{env(C)}$ by Def. 11 and $m' \in \Omega_{env(N)}$ by Def. 10. Thus, $m + m' \in \Omega_{N \oplus C}$ by Def. 3 and $m + m'$ is no deadlock of $N \oplus C$. $\square$

The minimal simulation relation of $beh_b(C)$ by $OG_b(N)$ can be computed in time and space proportional to $O(|beh_b(C)| \cdot |OG_b(N)|)$. Together with the annotation check, matching $beh_b(C)$ with $OG_b(N)$ has a complexity of $O(|beh_b(C)| \cdot |OG_b(N)| \cdot 2^{k+1})$, whereas $k = |I \uplus O|$ denotes the size of the interface. Consequently, checking whether an open net is a b-controller is decidable.

Theorem 13 (decidability of b-controllability). *Checking whether an open net is a b-controller of another open net= is decidable for every $b \in \mathbb{N}^+$.*

4 Accordance

An algorithm to decide accordance for two open nets *Spec* and *Impl* must decide whether every controller of *Spec* is also a controller of *Impl*. As an open net has potentially infinitely many controllers, we must check inclusion of two infinite sets. Because the set of all controllers of an open net can be represented in a finite manner using the operating guideline, we may use the operating guidelines of *Spec* and *Impl* to decide that *Impl* accords with *Spec*.

The b-accordance relation has been defined by Stahl and Vogler [8] and they showed that it is a precongruence for composition operator $\oplus$ and therefore supports compositional reasoning.

Definition 14 (b-accordance). Let $b \in \mathbb{N}^+$. For interface equivalent open nets *Impl* and *Spec*, *Impl b-accords with Spec*, denoted by $\mathit{Impl} \sqsubseteq^b_{acc} \mathit{Spec}$, if for all open nets C hold: C is a b-controller of *Spec* implies C is a b-controller of *Impl*.

We show that deciding accordance of *Impl* and *Spec* reduces to checking that the operating guideline of *Spec* simulates the operating guideline of *Impl* and that the corresponding formulae of related states imply each other.

Definition 15 (b-refinement). Let $b \in \mathbb{N}^+$. For interface equivalent open nets *Impl* and *Spec*, $OG_b(\mathit{Impl})$ *b-refines* $OG_b(\mathit{Spec})$, denoted by $OG_b(\mathit{Impl}) \sqsubseteq^b_{ref} OG_b(\mathit{Spec})$, if there exists a minimal simulation ϱ of $OG_b(\mathit{Spec})$ by $OG_b(\mathit{Impl})$ such that for each pair of nodes $(Q, Q') \in \varrho$:

1. $Q = Q_\emptyset$ implies $Q' = {Q_\emptyset}'$, and
2. the formula $\phi_{OG_b(\mathit{Spec})}(Q) \Rightarrow \phi_{OG_b(\mathit{Impl})}(Q')$ is a tautology.

The first item is crucial; otherwise, we could have a b-controller of *Spec* that is not a b-controller of *Impl* because it violates the bound only in the composition with *Impl* (the respective state is not reachable in the composition with *Spec*).

Consider Fig. 4. $OG_1(\mathit{Impl})$ 1-refines $OG_1(\mathit{Spec})$, but $OG_1(\mathit{Spec})$ does not 1-refine $OG_1(\mathit{Impl})$: Node $Q_\emptyset$ of $OG_1(\mathit{Impl})$ is related with node Q_1 of $OG_1(\mathit{Spec})$, thereby violating item (1) of Def. 15.

The next theorem justifies that refinement of operating guidelines and accordance coincide.

Theorem 16 (b-accordance vs. b-refinement). *Let $b \in \mathbb{N}^+$. For interface equivalent open nets Impl and Spec, $\mathit{Impl} \sqsubseteq^b_{acc} \mathit{Spec}$ iff $OG_b(\mathit{Impl}) \sqsubseteq^b_{ref} OG_b(\mathit{Spec})$.*

Proof. Let $OG_b(\mathit{Spec}) = (\mathcal{Q}, E, Q_0, \Sigma_{in}, \Sigma_{out}, \phi)$ and $OG_b(\mathit{Impl}) = (\mathcal{Q}', E', Q'_0, \Sigma_{in}, \Sigma_{out}, \phi')$ be the operating guidelines of open nets *Spec* and *Impl*, respectively.

($\Rightarrow$): Let $\mathit{Impl} \sqsubseteq^b_{acc} \mathit{Spec}$. Consider an open net C whose behavior $beh_b(C)$ is isomorph to the underlying automaton of $OG_b(\mathit{Spec})$ and that has a final state if literal *final* occurs in the annotation of the respective node. Clearly, C is a

b-controller of *Spec* and of *Impl*. Thus, by Definition 11, there exists a minimal simulation relation of $beh_b(C)$ by $OG_b(Impl)$, and hence there is a minimal simulation relation ϱ of $OG_b(Spec)$ by $OG_b(Impl)$.

Let $Q \in \mathcal{Q}$, and let β be an arbitrary assignment to literals occurring in $\phi(Q)$ with β evaluates $\phi(Q)$ to *true*. Remove from the underlying automaton of $OG_b(Spec)$ and node Q all outgoing, x-labeled edges where $\beta(Q)(x)$ is *false*. By Definition 11, the corresponding automaton still matches with *Spec* and thus with *Impl*. Let $Q' \in \mathcal{Q}'$ with $(Q, Q') \in \varrho$. Using Definition 11 again, we can see that β satisfies $\phi'(Q')$ as well. Thus, $\phi(Q) \Rightarrow \phi'(Q')$ is a tautology, for all $(Q, Q') \in \varrho$.

Assume now that $Q = Q_\emptyset$. A b-controller C of *Spec* could be in a marking m that violates bound b, and m is related with $Q_\emptyset$. By assumption, C is a b-controller of *Impl* and hence we conclude that for all $Q' \in \mathcal{Q}'$, $(Q_\emptyset, Q')$ in the simulation relation of $OG_b(Spec)$ by $OG_b(Impl)$ implies $Q' = {Q_\emptyset}'$ (as otherwise $Impl \oplus C$ is not b-bounded).

($\Leftarrow$): Let $OG_b(Impl) \sqsubseteq^b_{ref} OG_b(Spec)$ and C be a b-controller of *Spec*. We have to show that C is b-controller of *Impl*, too.

By Definition 11, there exists a minimal simulation relation $\varrho_{beh_b(C),OG_b(Spec)}$ of $beh_b(C)$ by $OG_b(Spec)$ and, by assumption, we also have a minimal simulation relation $\varrho_{OG_b(Spec),OG_b(Impl)}$ of $OG_b(Spec)$ by $OG_b(Impl)$. As simulation is transitive we conclude that $\varrho_{beh_b(C),OG_b(Impl)}$ is a simulation relation of $beh_b(C)$ by $OG_b(Impl)$. Relation $\varrho_{beh_b(C),OG_b(Impl)}$ is even a minimal simulation relation, because the underlying automata of $OG_b(Spec)$ and $OG_b(Impl)$ are deterministic by construction.

By assumption, $beh_b(C)$ matches with $OG_b(Spec)$; that is, for all markings m with $(m, Q) \in \varrho_{beh_b(C),OG_b(Spec)}$ and m is stable in $env(C)$, m satisfies $\phi(Q)$. In addition, we know $\phi(Q) \Rightarrow \phi'(Q')$, for all $(Q, Q') \in \varrho_{OG_b(Spec),OG_b(Impl)}$. Hence, m satisfies $\phi(Q')$, for all $(m, Q') \in \varrho_{beh_b(C),OG_b(Impl)}$.

Suppose there exists a marking m of C that is not b-bounded. Then, by Definition 11, for all $Q \in \mathcal{Q}$, $(m, Q) \in \varrho_{beh_b(C),OG_b(Spec)}$ implies $Q = Q_\emptyset$. By assumption, for each pair of nodes $(Q, Q') \in \varrho_{OG_b(Spec),OG_b(Impl)}$, $Q = Q_\emptyset$ implies $Q' = {Q_\emptyset}'$; thus, we conclude $(m, Q') \in \varrho_{beh_b(C),OG_b(Impl)}$ implies $Q' = {Q_\emptyset}'$. $\square$

We proceed with a short complexity analysis. Let $b \in \mathbb{N}^+$, and let *Impl* and *Spec* be interface equivalent open nets. A minimal simulation relation of $OG_b(Impl)$ by $OG_b(Spec)$ can be computed in time and space proportional to $O(|OG_b(Impl)| \cdot |OG_b(Spec)|)$. Let $k = |I \uplus O|$ denote the size of the interface. Then, checking whether *Impl* b-refines *Spec* has a complexity of $O(|OG_b(Impl)| \cdot |OG_b(Spec)| \cdot 2^{k+1})$. So checking b-accordance is decidable.

Theorem 17 (decidability of b-accordance). *Checking b-accordance of two open nets is decidable for every $b \in \mathbb{N}^+$.*

5 Conclusion

We have investigated the accordance precongruence of services. A service *Impl* accords with a service *Spec* if every controller of *Spec* (i.e., every service that deadlock freely communicates with *Spec*) is also a controller of *Impl*. We have presented a novel way to decide accordance. To this end, we used the notion of an operating guideline [4], which represents all controllers of a service in a finite manner. We have adapted the procedure of checking whether a service is a controller of an a given service and is, thus, contained in the operating guideline. In addition, we have also adapted the procedure for deciding accordance [7] for two services *Spec* and *Impl* based on their operating guidelines.

In contrast to [4], we considered controllers with unbounded interior. This caused the adaptation of the techniques introduced in [4,7], because we need to distinguish whether a controller can potentially violate the bound in the composition or not. The definition of matching (see Def. 11) extends the respective definition in [4] by item 2(a), where we require that states, in which the controller violates the bound, are not reachable in the composition. Similar, item (1) in the definition of operating guideline refinement (see Def. 15) extends the respective definition in [7]. Also here, we assign a more prominent role to the empty node: The new accordance check has to distinguish whether an input is enabled in the empty node or in another *true* annotated node—that is, whether the input is enabled in a reachable state or not.

In ongoing work, we aim to study efficient procedures to decide accordance for stricter termination criteria than deadlock freedom, including responsiveness [10] (i.e., controllers either terminate or have the possibility to communicate) and weak termination (i.e., the service has always the possibility to terminate).

References

1. Kaschner, K.: Conformance testing for asynchronously communicating services. In: ICSOC 2011. LNCS, vol. 7084, pp. 108–124. Springer (2011)
2. Liske, N., Lohmann, N., Stahl, C., Wolf, K.: Another approach to service instance migration. In: ICSOC 2009. pp. 607–621. LNCS 5900, Springer-Verlag (2009)
3. Lohmann, N.: Correcting deadlocking service choreographies using a simulation-based graph edit distance. In: BPM 2008. pp. 132–147. LNCS 5240, Springer-Verlag (2008)
4. Lohmann, N., Massuthe, P., Wolf, K.: Operating guidelines for finite-state services. In: ICATPN 2007. LNCS, vol. 4546, pp. 321–341. Springer (2007)
5. Lohmann, N., Wolf, K.: Compact representations and efficient algorithms for operating guidelines. Fundam. Inform. 107, 1–19 (2011)
6. Papazoglou, M.P.: Web Services: Principles and Technology. Pearson (2007)
7. Stahl, C., Massuthe, P., Bretschneider, J.: Deciding substitutability of services with operating guidelines. In: ToPNoC II. pp. 172–191. LNCS 5460, Springer (2009)
8. Stahl, C., Vogler, W.: A trace-based service semantics guaranteeing deadlock freedom. Acta Informatica 49(2), 69–103 (2012)
9. Vogler, W.: Modular Construction and Partial Order Semantics of Petri Nets, LNCS, vol. 625. Springer (1992)
10. Vogler, W., Stahl, C., Müller, R.: A trace-based semantics for responsiveness. In: ACSD 2012. IEEE Computer Society (2012), to appear