New Algorithms for Deciding the Siphon-Trap Property

Olivia Oanea, Harro Wimmel, and Karsten Wolf

Universität Rostock, Institut für Informatik, 18051 Rostock, Germany
{olivia.oanea,harro.wimmel,karsten.wolf}@uni-rostock.de

Abstract. The siphon-trap property, also known as Commoner-Hack property, establishes a relation between structural entities within a Petri net – the eponymous siphons and traps. The property is linked to the behavior of a Petri net, for instance to deadlock freedom and liveness of the net. It is nevertheless nontrivial to decide the property as a net can have exponentially many siphons and traps even if only minimal siphons are considered. Consequently, the value of the property depends on the availability of powerful decision procedures.

We contribute to this issue by proposing two new methods for deciding the siphon-trap property. One is a plain translation of the property into a Boolean satisfiability (SAT) problem, which exploits the fact that incredibly powerful SAT solvers are available. The second procedure has a divide-and-conquer nature which builds upon a decomposition of a Petri net into *open nets* and projects information about siphons and traps onto the interfaces of the components.

Keywords: Petri nets, Traps, Siphons, Commoner-Hack, Liveness, SAT, Divide-and-Conquer.

1 Introduction

The siphon-trap property [5,2] is a classical structural property of Petri nets. It states that every siphon (a set of places that cannot switch from unmarked to marked) includes a marked trap (a structure that cannot switch from marked to unmarked). The property can be used for deciding liveness in free choice Petri nets and as a sufficient condition for deadlock freedom in general Petri nets. According to common belief, the main advantage of structural techniques is that they avoid the generation of a state space which is subject to the state explosion problem. In fact, the siphon-trap property involves the investigation of only finitely many finite siphons in the net even for unbounded Petri nets, i.e. infinite state systems. Nevertheless, evaluating the property is far from trivial. Existing tools like INA [6] enumerate potentially exponentially many siphons and may thus run into severe run time and space problems.

We propose two new approaches for evaluating the siphon-trap property of place-transition nets. The first approach translates the property into a Boolean satisfiability problem. Our translation improves results in [10,1] where the property was translated into a Horn-satisfiability problem for bounded free-choice and

J. Lilius and W. Penczek (Eds.): PETRI NETS 2010, LNCS 6128, pp. 267–286, 2010.

other subclasses of Petri nets. The translation as such can be done in polynomial time resulting in a formula with $n(n+1)$ propositions, where n is the number of places. The subsequent satisfiability problem is NP-complete but there is a number of tools available which are capable of solving incredibly large instances in reasonable time.

The second approach follows the divide-and-conquer paradigm. We decompose a Petri net into *open net* components where an open net is a place bordered subnet such that each place on the border (we shall call them *interface places*) represents unidirectional asynchronous communication with exactly one other component. We improve an existing decomposition technique in two directions. First, we present a more efficient algorithm. Second, we propose a net transformation which allows us to divide a net into arbitrarily small components. For each component, information about siphons, traps, and their mutual relation is condensed into constraints for the interface places. Upon composition of components, information of the components is aggregated to corresponding information about the composite open net. Since the size of the condensed information has a stronger correlation to the number of *interface* places than to the overall number of places in a component, the approach has the potential of outperforming traditional algorithms at least for a significant class of nets. This in turn is sufficient for including an algorithm into a tool as present day computing environments support the parallel execution of several tasks.

2 Basic Definitions

Definition 1 (Petri net). *An* (unmarked) net *is a triple* (S, T, F) *where* S *and* T *are finite sets with* $S \cap T = \emptyset$, *and* F *is a mapping* $F\colon (S\times T)\cup(T\times S) \to \{0,1\}$, *i.e. we consider nets without arc weights.*

For any unmarked net (S, T, F) *and any* $x \in S \cup T$, *let* ${}^\bullet x := \{y \mid F(y,x) \neq 0\}$ *and* $x^\bullet := \{y \mid F(x,y) \neq 0\}$ *be the* preset *and* postset *of* x, *respectively. We extend this notion to sets* $X \subseteq S \cup T$ *by* ${}^\bullet X := \bigcup_{x\in X} {}^\bullet x$ *and* $X^\bullet := \bigcup_{x\in X} x^\bullet$. *We assume nets have no isolated places, i.e. places* s *with* ${}^\bullet s \cup s^\bullet = \emptyset$.

A marking *of* (S, T, F) *is a function* $m\colon S \to \mathbb{N}$. *We say that a place* s *has* k *tokens under* m *if* $m(s) = k$. *For* $S' \subseteq S$ *we introduce the abbreviation* $m(S') := \sum_{s\in S'} m(s)$ *and say that* S' *is* marked under m *iff* $m(S') > 0$, *otherwise it is* unmarked.

A marked net *is a tuple* (S, T, F, m_0) *consisting of an unmarked net* (S, T, F) *and an (initial) marking* m_0. *An* open net (S, T, F, m_0, S_i, S_o) *contains a marked net* (S, T, F, m_0), *a set* S_i *of* input *places with* $S_i \subseteq S$ *and* ${}^\bullet S_i = \emptyset$, *a set of* output *places* S_o *with* $S_o \subseteq S$ *and* $S_o{}^\bullet = \emptyset = S_i \cap S_o$. *The set* $I := S_i \cup S_o$ *is called the* interface *of the net, places in* $S\backslash I$ *are called* inner *places. Nets with an empty interface or without an interface at all are called* closed nets.

Open nets can be seen as partial nets mergeable via parts of their interfaces using a composition operator $\oplus$.

Definition 2 (Composition of open nets). *For* $k \in \{1, 2\}$ *let* $N_k = (S_k, T_k, F_k, m_k, S_{i,k}, S_{o,k})$ *be open nets such that* $T_1 \cap T_2 = \emptyset$, $S_{i,1} \cap S_{i,2} = \emptyset = S_{o,1} \cap S_{o,2}$, *and* $S_1 \cap S_2 = (S_{i,1} \cap S_{o,2}) \cup (S_{i,2} \cap S_{o,1})$, *i.e. common elements of the two open nets are non-inner places only, and these must be input in one and output in the other open net. Furthermore, for all* $s \in S_1 \cap S_2$: $m_1(s) = m_2(s)$ *must hold. Then we define* $N_1 \oplus N_2 := (S_1 \cup S_2, T_1 \cup T_2, F_1 \cup F_2, m_1 \cup m_2, S_i, S_o)$, *where* $S_i = (S_{i,1} \cup S_{i,2}) \backslash (S_1 \cap S_2)$ *and* $S_o = (S_{o,1} \cup S_{o,2}) \backslash (S_1 \cap S_2)$.

Note that $m_1 \cup m_2$ is well-defined since m_1 and m_2 are equal for common places. The composition $\oplus$ is obviously commutative. Associativity is also easy to see, we notice that a place may appear in the open nets of a well-defined expression of the form $N_1 \oplus N_2 \oplus N_3 \oplus \ldots$ either twice (once as input and once as output place, to be merged to one inner place) or once (as input, output or inner place) or not at all. Matching input and output places in different ways depending on the order of nets is therefore impossible and we can conclude:

Proposition 1. *The composition* $\oplus$ *is commutative and associative.*

Our main consideration are traps and siphons. A trap is a set of places that cannot be emptied once it contains a token, no matter which transitions fire. A siphon is a set of places that cannot obtain new tokens once it has been emptied of tokens.

Definition 3 (Traps and siphons). *A* trap Q *of an (unmarked or marked) net* (S, T, F, m_0) *is a set* $Q \subseteq S$ *with* $Q \neq \emptyset$ *and* $Q^\bullet \subseteq {}^\bullet Q$. *Analogously, a* siphon *is a set* $D \subseteq S$ *with* $D \neq \emptyset$ *and* ${}^\bullet D \subseteq D^\bullet$. *A trap* Q *is* marked *if* $\exists s \in S$: $m_0(s) > 0$. *For a set* X *of places of an open net* (S, T, F, m_0, S_i, S_o), *call* $I(X) = X \cap (S_i \cup S_o)$ *the* interface *of* X. *Let such a set be* closed *if* $I(X) = \emptyset$, *otherwise* open. *Let a siphon (or trap, resp.)* M *be* X-minimal *iff* $X \subseteq M$ *and no other siphon* D *(or trap, resp.) fulfills* $X \subseteq D \subset M$. *For a net* N *let* $\mathcal{Q}(N)$ *denote the set of all traps in* N *and* $\mathcal{D}(N)$ *the set of all siphons in* N.

A net $N = (S, T, F, \ldots)$ is called a *free-choice* net if for each pair $t, t' \in T$, ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$ implies ${}^\bullet t = {}^\bullet t'$. For these free-choice nets there is a well known relation between traps/siphons and liveness, i.e. whether all transitions can be enabled from all reachable markings.

Proposition 2 (Commoner-Hack [5,2]). *Let* N *be a marked free-choice net. Then* N *is live if and only if every siphon of* N *contains a marked trap.*

If we consider general nets we can only conclude:

Proposition 3. *Let* N *be a marked net.*

(1) If N *is live then every siphon of* N *contains a marked trap.*
(2) If every siphon of N *contains a marked trap then* N *does not contain deadlocks (i.e. all reachable markings enable at least one transition).*

In the sequel, we shall refer to the property "every siphon contains a marked trap" as the *siphon-trap property* (STP). The remainder of this article is devoted to new decision procedures for the property.

3 Evaluating the Siphon-Trap Property Using SAT

In this section we propose a reduction of STP to the famous SAT problem [3]. We aim at a formula that is satisfiable if and only if there is a siphon which does not contain a marked trap. Our starting point is a formula which operates on the places as propositions and whose satisfying assignments correspond exactly to the siphons of a given net. Such formula is well known.

Lemma 1 ([10,7]). *A set D of places of a net N is a siphon if and only if the assignment β with $\beta(s) = true$ if and only if $s \in D$ satisfies*

$$\bigvee_{s \in S} s \wedge \bigwedge_{t \in T} \bigwedge_{s \in t^\bullet} (s \Longrightarrow \bigvee_{s' \in {}^\bullet t} s').$$

The first part of the formula states the non-emptiness while the second part is the siphon condition ${}^\bullet D \subseteq D^\bullet$. A dual formula is capable of describing traps but can not immediately be used for formulating the STP. The reason is that there is a change of quantifiers: there *exists* a siphon D such that *every* included trap is unmarked. Hence we use another approach exploiting the fact that every siphon D containing traps has a unique maximal trap (which is the union of all traps included in D). Beginning with a siphon D, its maximal included trap can be computed by a repeated removal of places s where some post-transition has no post-place in the so far remaining set. Let n be the number of places in N. We represent the repetition of the procedure by introducing $(n + 1)$ variables $s^{(0)}, \dots, s^{(n)}$ for each place s. The variables $s^{(0)}$ represent a non-empty siphon as mentioned above. The variables $s^{(i)}$ represent intermediate stages D_i of the procedure for generating the maximal included trap. D_{i+1} is obtained from D_i by removing all places for which some post-transition does not have any post-place in D_i. Since there are only n places, the procedure converges after at most n iterations, so D_n is either empty or the maximal trap included in D. The relation between D_i and D_{i+1} can be expressed for each place s individually as follows:

$$s^{(i+1)} \iff (s^{(i)} \wedge \bigwedge_{t \in T} \bigwedge_{s \in {}^\bullet t} \bigvee_{s' \in t^\bullet} s'^{(i)}).$$

As we want to have the formula satisfied iff the maximal trap is unmarked or non-existent, we add the formula

$$\bigwedge_{s \in S : m_0(s) > 0} \neg s^{(n+1)}.$$

From these considerations, the following theorem is evident.

Theorem 1. *In a given net N with n places, there exists a siphon which does not include a marked trap if and only if the following formula is satisfiable:*

$$\begin{aligned} \phi ::= \quad & \bigvee_{s \in S} s^{(0)} \wedge \bigwedge_{t \in T} \bigwedge_{s \in t^\bullet} (s^{(0)} \Longrightarrow \bigvee_{s' \in {}^\bullet t} s'^{(0)}) & (1) \\ & \wedge \bigwedge_{i=0}^{n} \bigwedge_{s \in S} (s^{(i+1)} \iff (s^{(i)} \wedge \bigwedge_{t \in T} \bigwedge_{s \in {}^\bullet t} \bigvee_{s \in t^\bullet} s^{(i)})) & (2) \\ & \wedge \bigwedge_{s \in S : m_0(s) > 0} \neg s^{(n+1)} & (3) \end{aligned}$$

Table 1. Evaluating STP: SAT vs. INA

ID	$\|P\|$	$\|T\|$	$\|F\|$	SAT	INA
phils10	50	40	120	0.05 sec	3 sec
phils20	100	80	240	0.24 sec	≥2h
phils50	250	200	600	2.29 sec	n.a.
phils100	500	400	1200	12 sec	n.a.
phils150	750	600	1800	40 sec	n.a.
phils200	1000	800	2400	119 sec	n.a.
data1010	50	40	300	0.12 sec	8 sec
data1212	60	48	408	0.19 sec	16 sec
data1515	75	60	600	0.36 sec	28 sec

The formula contains $n(n+1)$ different propositions, one for each place and iterative step (counted by t), and has obviously a length that is polynomial in $card(S) + card(T) + card(F)$.

We have implemented an ad-hoc translation from a Petri net to the mentioned formula and shipped it to the state-of-the-art SAT checker MiniSat [9] and compared our results with the STP check done by INA [6]. We obtained the results listed in Table 1. As experimental data, we used the k dining philosophers examples and the semaphore based scheme for concurrent read and exclusive write access to a database with k writing and k reading processes. Observe that the INA check time explodes for the 20 philosophers example while the SAT check has a significant time increase for the 200 philosophers example.

4 Evaluating the Siphon-Trap Property Using a Divide-and-Conquer Approach

Deciding liveness is co-NP-complete for free-choice nets according to Esparza and Nielsen [4], so a general fast algorithm is impossible. In the following, we develop an algorithm for evaluating the STP using a divide-and-conquer strategy. The complexity of this algorithm depends more on the size of interfaces during the conquer part than on the size of the nets. Managing to keep the interfaces small may thus lead to a fast algorithm. The general algorithm will look like this:

1. Decompose a (marked) net $N = (S, T, F, m_0)$ into a set of open net components.
2. Calculate traps and siphons for each such component. For closed siphons, the STP is evaluated using any traditional algorithm.
3. Condense information about open siphons and included traps such that it only refers to the interface.
4. Aggregate components step by step. From the information provided by the components, reason about siphons that become closed through the aggregation and derive information about open siphons and included traps of the aggregated open net.

In Subsection 4.1, we propose a procedure that is able to decompose a Petri net into arbitrarily small open nets. How far to break down a Petri net is optional though. Subsection 4.2 studies the relations between siphons and traps on one hand and open net composition on the other. In Subsection 4.3 we define a structure that is later on used for representing the information about open siphons and traps. Then, we take this information for reasoning about siphons and traps that are closed by aggregation. Finally, we deal with the generation of information about open siphons and traps in an aggregation.

4.1 Decomposition into Open Nets

So far we have talked about some aspects of components but we have not defined them yet. Thanks to the composition $\oplus$, this is easy to do.

Definition 4 (Components). *Let N be a marked net and N_1 be an open net. We call N_1 a* component *of N if there is some open net N_2 with $N = N_1 \oplus N_2$.*

For our divide-and-conquer approach we are usually interested in small components, i.e. we would like to split a net into as many components as possible. Zaitsev [11] presented an algorithm to obtain the unique set of smallest components into which a net can be decomposed. Later, the algorithm was improved by Mennicke et al. [8]. Its idea is to start at some transition and recursively tag necessary net elements until a component is completed:

Definition 5 (Building components). *Let $N = (S, T, F, m_0, S_i, S_o)$ be an open net and $t \in T$ a transition. The* component *$C(t) = (S', T', F|_{(S' \times T' \cup T' \times S')}$, $m_0|_{S'}$, S'_i, $S'_o)$ is the smallest (wrt. set inclusion) open net fulfilling the following criteria:*

(1) $t \in T'$,
(2) if $t' \in T'$ then ${}^\bullet t' \cup t'^\bullet \subseteq S'$,
(3) if $t' \in T'$ then $({}^\bullet t')^\bullet \cup {}^\bullet(t'^\bullet) \subseteq T'$,
(4) for $s \in S'$: $(s \in S_i \vee \exists t' \in T \backslash T' : t' \in {}^\bullet s) \Longrightarrow s \in S'_i$,
(5) for $s \in S'$: $(s \in S_o \vee \exists t' \in T \backslash T' : t' \in s^\bullet) \Longrightarrow s \in S'_o$.

Any open net can be disassembled into a set of at most $|T|$ different components (one for each transition, but $t' \in C(t)$ implies $C(t) = C(t')$). Different components have disjoint sets of transitions and inner places. Interface places may be shared by components, but each such place may appear only once as input place and once as output place in all components together. Clearly, $\bigoplus_t C(t) = N$ if we add only one of $C(t)$, $C(t')$ whenever $C(t) = C(t')$.

Example. There are nets which can be split up into components with only one transition in each. Take e.g. a cycle of alternating places and transitions, with two places before and after each transition, and one transition before and after each place. All components look alike, the first two being N_1 and N_2 of Fig. 1. Composing further components to the resulting net $N_1 \oplus N_2$ may prolong the strand until the final component with output places a and b is added to complete the cycle.

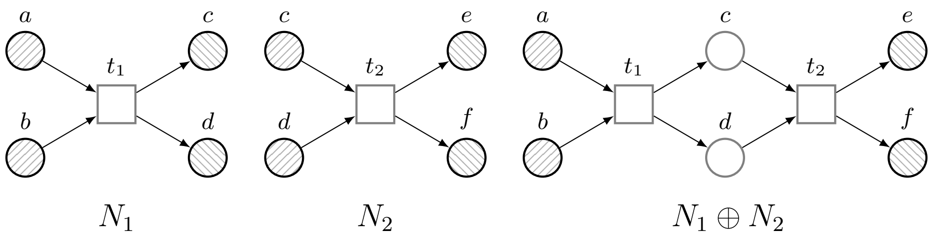

Fig. 1. Two components N_1 and N_2 and their composition $N_1 \oplus N_2$. Input places have stripes going upwards, output places downwards.

Since the components are so small, we can easily determine all traps and siphons, e.g. for N_1: $\mathcal{Q} = \{\{c\}, \{d\}, \{c,d\}, \{a,c\}, \{b,c\}, \{a,b,c\}, \{a,d\}, \{b,d\}, \{a,b,d\}, \{a,c,d\}, \{b,c,d\}, \{a,b,c,d\}\}$ and $\mathcal{D} = \{\{a\}, \{b\}, \{a,b\}, \{a,c\}, \{a,d\}, \{a,c,d\}, \{b,c\}, \{b,d\}, \{b,c,d\}, \{a,b,c\}, \{a,b,d\}, \{a,b,c,d\}\}$.

If we restrict ourselves e.g. to $\{s\}$-minimal traps and siphons for some place $s \in S$, we get the even smaller sets $\mathcal{Q}_1 = \{\{c\}, \{d\}, \{a,c\}, \{b,c\}, \{a,d\}, \{b,d\}\}$ and $\mathcal{D}_1 = \{\{a\}, \{b\}, \{a,c\}, \{a,d\}, \{b,c\}, \{b,d\}\}$.

The conquer part of our divide-and-conquer strategy should later show the siphons and traps of $N_1 \oplus N_2$ to be $\mathcal{Q}' = \{\{e\}, \{f\}, \{c,e\}, \{d,e\}, \{c,f\}, \{d,f\}, \{a,c,e\}, \{b,c,e\}, \{a,d,e\}, \{b,d,e\}, \{a,c,f\}, \{b,c,f\}, \{a,d,f\}, \{b,d,f\}\}$ and $\mathcal{D}' = \{\{a\}, \{b\}, \{a,c\}, \{a,d\}, \{b,c\}, \{b,d\}, \{a,c,e\}, \{a,c,f\}, \{a,d,e\}, \{a,d,f\}, \{b,c,e\}, \{b,c,f\}, \{b,d,e\}, \{b,d,f\}\}$ (again with the reduction to s-minimal elements for $s \in S$). □

Size reduction of components. If the components are not as small as those in Fig.1 we might like to split them up even more as the number of siphons and traps of a component may grow exponentially with its size, i.e. the number of places. Two transitions with a common place in either their presets or in their postsets always belong to the same component. To force them to different components we need to split up the place *before* we dissolve the net into its components. We propose the following operation, which will replace one place by a circle of alternating places and transitions.

Definition 6 (Replacing places). *Let $N = (S, T, F, m_0, S_i, S_o)$ be an open net and p an inner place of S. Take any partition $P = \{T_i \mid 1 \leq i \leq n\}$ of ${}^\bullet p \cup p^\bullet$ where n is the number of the sets T_i in P. We define $N(p, P) := (S', T', F', m'_0, S_i, S_o)$ by the following algorithm:*

- *Start with $S' = S \backslash \{p\}$, $T' = T$, $F' = F|_{(S' \times T) \cup (T \times S')}$ and $m'_0 = m_0|_{S'}$.*
- *For each T_i add a place p_i and for each $t \in T_i$ connect it like p: $F'(p_i, t) = F(p, t)$ and $F'(t, p_i) = F(t, p)$.*
- *If exists $t \in T_i$ with $F'(t, p_i) > 0$ add p_i^e and t_i^e with $F'(p_i^e, t_i^e) = F'(t_i^e, p_i) = 1$.*
- *If exists $t \in T_i$ with $F'(p_i, t) > 0$ add p_i^x and t_i^x with $F'(p_i, t_i^x) = F'(t_i^x, p_i^x) = 1$.*
- *For each $i \in \{1, \ldots, n\}$ identify the last existing place of the list p_i^e, p_i, p_i^x with the first one of the list $p^e_{(i \bmod n)+1}$, $p_{(i \bmod n)+1}$, $p^x_{(i \bmod n)+1}$, forming a circle of all the newly added places and transitions.*

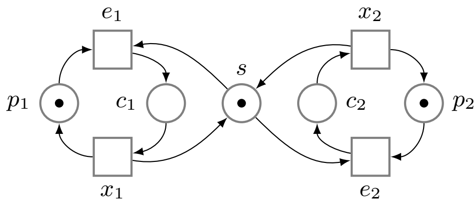

Fig. 2. A semaphore net N. For the two processes p_1 and p_2 on the left and right, transitions e and x mean entry to and exit from the critical section c, the semaphore is place s. Note that N has an empty interface.

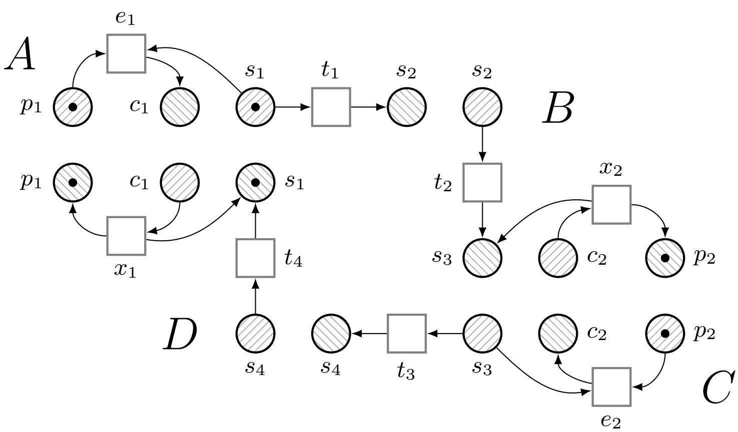

Fig. 3. The semaphore place s has been replaced by a circle (consisting of the s_i and t_i). The semaphore net dissolves into four components A, B, C, and D, where places to be identified when rejoining the components have been given the same label.

– *Set* $m'_0(p_1) = m_0(p)$ *and* $m'_0(s) = 0$ *for all other places on the newly formed circle.*

Example. Consider the semaphore net of Fig. 2 with the two processes p_1-e_1-c_1-x_1 and p_2-e_2-c_2-x_2 being in their critical section at c_1 and c_2, respectively, and the semaphore place s. The net only has two components, one with the transitions e_1 and e_2, the other with x_1 and x_2. We cannot split it along the process boundaries, as both processes need read and write access to the place s.

If we replace s by a circle of four places and transitions, we obtain four components A, B, C, and D as shown in Fig. 3. It becomes possible now to merge components such that we get subnets $A \oplus D$ and $B \oplus C$ consisting of one full process each. These compositions have the smallest number of traps and siphons of all combinations of two components, which reduces time and space needed for the conquer part of our algorithm. Accidentally (or not), these are also the compositions with the smallest interfaces. □

The question is now what will happen to the traps and siphons if we replace a place by a complete circle. We find the nice property that traps and siphons are bijectively mapped between the two nets.

Proposition 4 (Unchanged traps and siphons). *Let N be an open net with an inner place p and P a partition of $^\bullet p \cup p^\bullet$. For $N(p, P)$ according to definition 6 let r be a map with $r(p_i^e) = r(p_i) = r(p_i^x) = p$ for all places added in the construction of $N(p, P)$ and $r(s) = s$ for all places s of N except p. Then, for all subsets X of places of N: X is a trap of N iff $r^{-1}(X)$ is a trap of $N(p, P)$ and X is a siphon of N iff $r^{-1}(X)$ is a siphon of $N(p, P)$. Furthermore, all traps and siphons of $N(p, P)$ have the form $r^{-1}(X)$.*

Proof. We show this for traps only; for siphons the proposition then follows from symmetry. Let $r^{-1}(X) \in \mathcal{Q}(N(p, P))$ and $t \in X^\bullet$ a transition of N. Then, t is also a transition in $N(p, P)$ and $t \in r^{-1}(X)^\bullet$ by Def. 6. As $r^{-1}(X)$ is a trap, $t \in {}^\bullet r^{-1}(X)$ in $N(p, P)$ and therefore also $t \in {}^\bullet X$ in N. We conclude $X \in \mathcal{Q}(N)$. The same argument holds for $X \in \mathcal{Q}(N)$ and $t \in r^{-1}(X)^\bullet$ in $N(p, P)$ if we just swap X with $r^{-1}(X)$ and N with $N(p, P)$. We conclude $r^{-1}(X) \in \mathcal{Q}(N(p, P))$ then.

Let now Y be a trap of $N(p, P)$. If Y does not contain any of the $p_i^e/p_i/p_i^x$, then $p \notin r(Y)$ and r is the identity on Y. We conclude $Y = r^{-1}(r(Y))$. If Y contains at least one of the $p_i^e/p_i/p_i^x$ we get $p \in r(Y)$. By the trap property, if a $t_i^e \in p_i^{e\bullet}$ for some $p_i^e \in Y$, also $t_i^e \in {}^\bullet Y$ must hold, i.e. $p_i \in Y$. Analogously, for $t_i^x \in p_i{}^\bullet$ with $p_i \in Y$ also $t_i^{x\bullet} = \{p_i^x\} \subseteq Y$ holds. In any case, if one of the $p_i^e/p_i/p_i^x$ belongs to Y, all of them do for all i. So again, $Y = r^{-1}(r(Y))$. □

Note that Def. 6 cannot be applied to interface places. This would change the number of siphons and traps in the net, as the circle constructed in the definition cannot contain interface places. Logically, the best time to apply Def. 6 is then at the beginning, when we usually have a closed net and could replace *all* the places. Then, components would all look like the one depicted in Fig. 4, where for each place in the preset or postset of the main transition t one link of the corresponding circle created by Def. 6 is added.

Not all of the sets of traps and siphons in figure 4 need to be considered for our divide-and-conquer approach, since Prop. 4 tells us that the circles of Def. 6 appear either completely or not at all in any trap or siphon of the whole net. That means, only the sets Q_2, D_2, Q_7, D_7, $Q_4 \times (Q_2 \cup Q_7)$, and $D_4 \times (D_2 \cup D_7)$ and unions of two or more traps or two or more siphons from these sets will be relevant subsets of traps and siphons of the overall net.

4.2 Composing Siphons and Traps

There is a good reason for using open net decomposition rather than any other style of decomposition.

Lemma 2. *Let N_1 and N_2 be open nets with $N_k = (S_k, T_k, F_k, m_{0,k}, S_{i,k}, S_{o,k})$ for $k = 1, 2$.*

- *If D is a siphon (or trap, resp.) in $N_1 \oplus N_2$ then $D \cap S_1$ is either empty or a siphon (or trap, resp.) in N_1 and $D \cap S_2$ is either empty or a siphon (or trap, resp.) in N_2.*

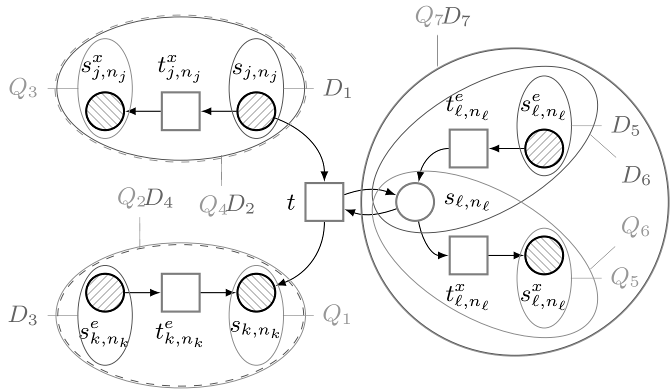

Fig. 4. A component $C(t)$ for some transition t with ${}^\bullet t = \{s_j, s_\ell\}$ and $t^\bullet = \{s_k, s_\ell\}$ as given by definition 6. Ellipses show the traps and siphons. The dashed ellipses Q_4 and D_4 are not traps or siphons (due to t) and need to be unified with traps from $Q_1/Q_2/Q_6/Q_7$ and siphons from $D_1/D_2/D_6/D_7$ first, respectively.

– *If D_1 is a siphon (or trap, resp.) in N_1 and D_2 is a siphon (or trap, resp.) in N_2 such that $D_1 \cap S_2 = D_2 \cap S_1$ (i.e. their interfaces to the respective other component are equal) then $D_1 \cup D_2$ is a siphon (or trap, resp.) in $N_1 \oplus N_2$.*

Proof. This follows easily from the constraints on interface places in open nets. Empty sets occur if D lies completely in the inner part of either N_1 or N_2. $\square$

Example. In Fig. 1, $\{a, d, e\}$ is a siphon and a trap of $N_1 \oplus N_2$. It decomposes into the siphons (and traps) $\{a, d\}$ of N_1 and $\{d, e\}$ of N_2. The other way round, $\{a, c\}$ is a siphon of N_1, $\{c\}$ is a siphon of N_2. c and d are the shared places of the interfaces of N_1 and N_2. Hence $\{a, c\}$ is a siphon in $N_1 \oplus N_2$. $\square$

From Lemma 2, the general idea of our approach is obvious. We collect, for each part of the interface of a component, the open siphons and the included traps, together with their interface. Upon composition, we merge siphons and traps with equal interface. Unfortunately, given an interface with k places, there are 2^k potential interfaces for siphons to be considered, and for each siphon, a contained trap can have an interface that spans over any subset of the interface of the siphon. Consequently, we need to further investigate regularities that arise from the open net shape of the components. To this end, we shall heavily exploit the following simple observations on siphons and traps.

Proposition 5 (Properties of Siphons)
(1) The union of siphons is a siphon.
(2) Let D be a siphon and $X \subseteq D$. There is an X-minimal siphon $D' \subseteq D$.
(3) Let D be a $\emptyset$-minimal siphon in $N_1 \oplus N_2$. Then, if not empty, $D \cap S_1$ is a $(D \cap S_1 \cap S_2)$-minimal siphon in N_1 and $D \cap S_2$ is a $(D \cap S_1 \cap S_2)$-minimal siphon in N_2.
The same observations hold for traps.

Proof. (1) and (2) are trivial. For (3) the places $D \cap S_1 \cap S_2$ are forced in the siphons while the remaining places follow by the same reasoning as for D in $N_1 \oplus N_2$, i.e. the structure of the net.

Let us first reduce the number of siphons to be considered. Consider two components N_1 and N_2 and a set of shared places $X \subseteq S_1 \cap S_2$. By Lemma 2, for *every* pair of siphons D_1 of N_1 and D_2 of N_2 where $D_1 \cap S_2 = X = D_2 \cap S_1$, $D_1 \cup D_2$ is a siphon in $N_1 \oplus N_2$. However, some of these siphons may contain more or better (i.e. marked) traps than others.

Definition 7 (Worse siphons). *Let N be an open net and $X \subseteq S_i \cup S_o$. Let D_1 and D_2 be siphons with $D_1 \cap (S_i \cup S_o) = X = D_2 \cap (S_i \cup S_o)$. Call D_1* worse *than D_2 iff, for every $Y \subseteq X$,*

- *If D_1 contains a trap Q_1 with $Q_1 \cap X = Y$ then D_2 contains a trap Q_2 with $Q_2 \cap X = Y$.*
- *If D_1 contains a* marked *trap Q_1 with $Q_1 \cap X = Y$ then D_2 contains a* marked *trap Q_2 with $Q_2 \cap X = Y$.*

Example. In $N_1 \oplus N_2$ of Fig. 1, siphons $\{a, c, e\}$ and $\{a, d, e\}$ are mutually worse than each other, so only one of them has to be considered in larger compositions. Assuming a token on d, $\{a, c, e\}$ is worse than $\{a, d, e\}$ but not vice versa. □

Lemma 3. *Let N_1 and N_2 be open nets. Let D_1 be a siphon of N_1 and let D_2 be a siphon of N_2 such that $D_1 \cap S_2 = D_2 \cap S_1$. Let D_1 be worse than D_1' and D_2 be worse than D_2'. Then $D_1 \cup D_2$ is worse than $D_1' \cup D_2'$ in $N_1 \oplus N_2$.*

In particular, if the union of the worse siphons includes a marked trap, so does the union of the better siphons. Consequently, we may remove a siphon from any consideration in a component as long as we keep a worse one. Although *worse than* is only a preorder and no partial order, we shall sloppily refer to the *worst* siphons as a (as small as possible) set of siphons that needs to be kept according to Lemma 3. The next observation rephrases the well-known fact that it is sufficient to check $\emptyset$*-minimal* siphons for evaluating the STP.

Corollary 1. *Let D_1 and D_2 be siphons of an open net N with $D_1 \cap (S_i \cup S_o) = D_2 \cap (S_i \cup S_o)$. If $D_1 \subseteq D_2$ then D_1 is worse than D_2.*

While the previous result reduces the number of siphons to be considered for a given interface, the following investigations concern the number of different interfaces to be explicitly considered. We shall argue, that finally we only need to consider *elementary* siphons and traps.

Definition 8 (Elementary siphons and traps). *A siphon D of an open net N is* elementary *iff there is a place $s \in S_i \cup S_o$ such that D is $\{s\}$-minimal. A trap Q is* interface-elementary *iff there is a place $s \in S_i \cup S_o$ such that Q is $\{s\}$-minimal. Q is* token-elementary *iff there is a place s where $m_0(s) > 0$ and Q is $\{s\}$-minimal.*

Example. In the open net N_1 of Fig. 1, $\{a\}, \{b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}$ are the elementary siphons. Although $\{a\}$ is included in $\{a, c\}$, we want to keep both as $\{a, c\}$ is $\{c\}$-minimal while $\{a\}$ is not. The interface-elementary traps are $\{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c\}, \{d\}$. There are no token-elementary traps. Assuming a token on a, $\{a, c\}, \{a, d\}$ would become token-elementary. Assuming instead a token on c, the only token-elementary trap would be $\{c\}$. In particular, the definition states that $\{b, c\}$ is not a token-elementary trap. □

Note that a token-elementary trap may be closed (i.e. disjoint to the interface) while an interface-elementary trap is always open. The following facts justify this selection.

Lemma 4. *Let N be an open net.*
(1) For every open siphon D of N there is a worse union of elementary siphons.
(2) If a siphon D contains a trap Q then it contains some union of interface-elementary traps $Q_1 \cup \ldots \cup Q_k$ where $Q \cap (S_i \cup S_o) = (Q_1 \cup \ldots \cup Q_k) \cap (S_i \cup S_o)$.
(3) If a siphon D contains a marked trap Q then it contains some union of traps $Q_1 \cup \ldots \cup Q_k \cup Q_m$ where $Q \cap (S_i \cup S_o) = (Q_1 \cup \ldots \cup Q_k \cup Q_m) \cap (S_i \cup S_o)$, $Q_1, \ldots, Q_k$ are interface-elementary, and Q_m is token-elementary.

Proof. (1) Let X $= D \cap (S_i \cup S_o) \neq \emptyset$. For each $s \in X$, let D_s be an $\{s\}$-elementary siphon included in D. Obviously, $\bigcup_{s \in X} D_s$ has the same interface as D, is contained in D, and not empty. By Cor. 1, it is worse than D. Claims (2) and (3) can be proven analogously, but note that $k = 0$ holds in the unions if Q is closed. □

In consequence, we only need to store information about elementary siphons, elementary traps, and information about inclusion of elementary traps in unions of elementary siphons. The advantage of using elementary traps and siphons is their simple structure. The following is trivial.

Lemma 5. *Let N be an open net.*
(1) For $s \in S_i$, $\{s\}$ is the only $\{s\}$-minimal siphon of N. For $s \in S_o$, $\{s\}$ is the only $\{s\}$-minimal trap of N.
(2) For $s \in S_o$ and an $\{s\}$-minimal siphon D, $D \cap S_o = \{s\}$. For $s \in S_i$ and an $\{s\}$-minimal trap Q, $Q \cap S_i = \{s\}$.

Definition 9 (Wrapping siphons). *A family $\mathcal{M} = \{D_1, \ldots, D_k\}$ of sets of places* wraps *a set Q of places iff $Q \subseteq D_1 \cup \ldots \cup D_k$ and this is not the case for any proper subset of $\mathcal{M}$.*

Example. In the net $N_1 \oplus N_2$ of Fig. 1, the family of siphons $\{\{a, c\}, \{b, d, e\}\}$ wraps the trap $\{c, e\}$. □

Remark 1. Let $\mathcal{M}$ be a family of elementary siphons. The union of $\mathcal{M}$ includes a trap Q if and only if Q is wrapped by some subset of $\mathcal{M}$.

Even among the elementary siphons, some siphon D may be redundant. This is the case if, for all siphons that can be constructed using D, a worse one can be constructed without using D.

Definition 10 (Redundant elementary siphon). *Let N be an open net and $\mathcal{M}$ a set of elementary siphons. Siphon $D \in \mathcal{M}$ is* redundant *iff, for all $\mathcal{M}_1 \subseteq \mathcal{M}$ there exists another subset $\mathcal{M}_2 \subseteq \mathcal{M}$ where $\bigcup(\mathcal{M}_2 \setminus \{D\})$ is worse than $\bigcup(\mathcal{M}_1 \cup \{D\})$. ($\bigcup X$ without a subscript stands for $\bigcup_{x \in X} x$.)*

Example. In $N_1 \oplus N_2$ of Fig.1, any of the elementary siphons $\{a, c, e\}$ and $\{a, d, e\}$ is redundant. In fact, any interface constellation of siphons and traps that can be composed from elementary objects and $\{a, c, e\}$ can as well be generated using $\{a, d, e\}$. After removing one of them, the other one is no longer redundant as it is then the only one remaining with interface $\{a, e\}$. If we put a token on d, only $\{a, d, e\}$ is redundant. For any constellation of siphons and included traps that can be constructed using $\{a, d, e\}$, a worse one (particularly with some unmarked traps instead of marked traps) can be generated using $\{a, c, e\}$. □

4.3 Representing Information about Open Siphons and Traps

From the considerations of the previous subsection, we conclude that we need to provide the following information about an open net.

Definition 11 (Information about components). *Let N be an open net, $\mathcal{M}_D$ a set of elementary siphons that can be obtained from the set of all elementary siphons by removing (one by one) redundant ones, $\mathcal{M}_Q$ the set of interface-elementary traps in N, and $\mathcal{M}_M$ the set of all token-elementary traps in N. Fix a set Σ with elements from an arbitrary universe such that $card(\Sigma) = card(\mathcal{M}_D)$ and fix some bijection l between Σ and $\mathcal{M}_D$ (elements of Σ serve as* names *for elementary siphons). We keep track of the following information about N:*

- *The set Σ introducing names for elementary siphons;*
- *A mapping $int : \Sigma \to \wp(S_i \cup S_o)$, $x \mapsto l(x) \cap (S_i \cup S_o)$ recording the interfaces of the elementary siphons;*
- *The set $L_Q = \{Q \cap (S_i \cup S_o) \mid Q \in \mathcal{M}_Q\}$ introducing the interfaces of the interface-elementary traps;*
- *The set $L_M = \{Q \cap (S_i \cup S_o) \mid Q \in \mathcal{M}_M\}$ introducing the interfaces of the token-elementary traps;*
- *The mapping $w_Q : L_Q \to \wp(\wp(\Sigma))$, $X \mapsto \{l^{-1}(\mathcal{M}) \mid \exists Q \in \mathcal{M}_Q : Q \cap (S_i \cup S_o) = X, \mathcal{M}$ wraps $Q\}$ recording the wrapping sets of elementary siphons for all interface-elementary traps with a given interface;*
- *The mapping $w_M : L_M \to \wp(\wp(\Sigma))$, $X \mapsto \{l^{-1}(\mathcal{M}) \mid \exists Q \in \mathcal{M}_M : Q \cap (S_i \cup S_o) = X, \mathcal{M}$ wraps $Q\}$ recording the wrapping sets of elementary siphons for all token-elementary traps with a given interface;*

Example. The full information about N_1 in Fig. 1 reads as follows.

- $\Sigma_1 = \{1, 2, 3, 4, 5, 6\}$;
- $int_1(1) = \{a\}, int_1(2) = \{b\}, int_1(3) = \{a, c\}, int_1(4) = \{a, d\}, int_1(5) = \{b, c\}, int_1(6) = \{b, d\}$;
- $L_{Q1} = \{\{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c\}, \{d\}\}$;
- $L_{M1} = \emptyset$;

- $w_{Q1}(\{a,c\}) = \{\{3\}\}, w_{Q1}(\{a,d\}) = \{\{4\}\}, w_{Q1}(\{b,c\}) = \{\{5\}\}$, $w_{Q1}(\{b,d\}) = \{\{6\}\}, w_{Q1}(\{c\}) = \{\{3\},\{5\}\}, w_{Q1}(\{d\}) = \{\{4\},\{6\}\}$;
- $w_M = \emptyset$.

Assuming a token on c, we would obtain $L_M = \{\{c\}\}$ and $w_M(\{c\}) = \{\{3\},\{5\}\}$. With a token on b instead, we would get $L_M = \{\{b,c\},\{b,d\}\}$, $w_M(\{b,c\}) = \{\{5\}\}$, and $w_M(\{b,d\}) = \{\{6\}\}$. For later use, we provide the full information for N_2 although it does not provide new insights.

- $\Sigma_2 = \{7,8,9,10,11,12\}$;
- $int_2(7) = \{c\}, int_2(8) = \{d\}, int_2(9) = \{c,e\}, int_2(10) = \{c,f\}, int_2(11) = \{d,e\}, int_2(12) = \{d,f\}$;
- $L_{Q2} = \{\{c,e\},\{c,f\},\{d,e\},\{d,f\},\{e\},\{f\}\}$;
- $L_{M2} = \emptyset$;
- $w_{Q2}(\{c,e\}) = \{\{9\}\}, w_{Q2}(\{c,f\}) = \{\{10\}\}, w_{Q2}(\{d,e\}) = \{\{11\}\}$, $w_{Q2}(\{d,f\}) = \{\{12\}\}, w_{Q2}(\{e\}) = \{\{9\},\{11\}\}, w_{Q2}(\{f\}) = \{\{10\},\{12\}\}$;
- $w_M = \emptyset$. □

In the remainder of this section, we argue that this information for some open nets N_1 and N_2 is sufficient for reasoning about siphons and traps of $N_1 \oplus N_2$. Let us first consider closed siphons in $N_1 \oplus N_2$. If a closed siphon is already a closed one in either N_1 or N_2, we assume that this siphon has been checked for elementary components, or has been checked during an earlier composition step. It is thus sufficient to consider those siphons D that spread over both N_1 and N_2. By the considerations in the previous subsection, it is sufficient to check those siphons for included marked traps which can be composed by elements of $\mathcal{M}_D$. Concerning the included traps, it is sufficient to check traps that can be composed by elements of $\mathcal{M}_Q$ and a single element of $\mathcal{M}_M$. We propose to execute the necessary checks simultaneously for all siphons by translating the check into a Boolean formula. The formula is satisfied if and only if some siphon of $N_1 \oplus N_2$ that spreads over both components does not contain a marked trap. The propositions of the formula are elements of Σ_1 and Σ_2, i.e. the symbols representing the elementary siphons of the two components (which we silently assume to be disjoint). The satisfying assignment assigns true to the names of those elementary siphons whose composition is a siphon that proves STP not to hold.

The formula consists of three parts. In the first part, we state that the represented siphon is not empty. In the second part, we state that the projections of the siphon to the components generate the same interface. In the third part, we state that the composition does not include a marked trap. The trick for stating the third part is to state that the siphon represented by the satisfying assignment does not include any wrap for at least one elementary trap participating in a trap of the composed system. Traps in the composed system are formed by a union of traps of the components such that the union of elementary traps in N_1 have the same interface to N_2 which the union of elementary traps of N_2 has to N_1. The following definition boils this idea down to interface considerations. As the same technique is later on needed for siphons as well, we already present matching for siphons as well.

Definition 12 (Matching). *Let N_1 and N_2 be open nets with information attached according to Def. 11. A* token trap matching *is a tuple* $[X_1, Y_1, X_2, Y_2]$ *such that* $X_1 \subseteq L_{M1}$, $Y_1 \subseteq L_{Q_1}$, $X_2 \subseteq L_{M2}$, $Y_2 \subseteq L_{Q_2}$, $card(X_1) + card(X_2) = 1$, $\bigcup(X_1 \cup Y_1) = \bigcup(X_2 \cup Y_2)$. *An* interface trap matching *is a tuple* $[Y_1, Y_2]$ *such that* $Y_1 \subseteq L_{Q_1}$, $Y_2 \subseteq L_{Q_2}$, $\bigcup Y_1 = \bigcup Y_2$. *A* siphon matching *is a tuple* $[Z_1, Z_2]$ *such that* $Z_1 \subseteq \Sigma_1$, $Z_2 \subseteq \Sigma_2$, *and* $\bigcup_{\sigma_1 \in Z_1} int(\sigma_1) = \bigcup_{\sigma_2 \in Z_2} int(\sigma_2)$. *A matching is minimal iff no different matching is pointwise set-included. A token trap matching is* internal *iff* $\bigcup X_1 \cup \bigcup Y_1 \subseteq S_2$ *and* $\bigcup X_2 \cup \bigcup Y_2 \subseteq S_1$. *The* interface *of a trap matching is* $(\bigcup(X_1 \cup Y_1 \cup X_2 \cup Y_2)) \backslash (S_1 \cap S_2)$ *resp.* $(\bigcup(Y_1 \cup Y_2)) \backslash (S_1 \cap S_2)$.

Example. There are no token-minimal matchings for N_1 and N_2 in Fig. 1. Examples of minimal siphon matchings are $[\{1\}, \emptyset]$, $[\{3\}, \{7\}]$, or $[\{3\}, \{9\}]$. Examples for minimal trap matchings are $[\emptyset, \{\{e\}\}]$ or $[\{\{c\}\}, \{\{c, e\}\}]$. Assuming a token on c in both components, $[\{\{c\}\}, \emptyset, \emptyset, \{\{c, e\}\}]$ would be a minimal token trap matching. □

Minimal matchings can be easily determined by a saturation algorithm. Start with an individual element. That may lead to interface places s that are not in the respective other open net. Add (nondeterministically) an $\{s\}$-minimal object of the other component and proceed until all interface places are matched. If there is no $\{s\}$-minimal object, just backtrack.

The definition shows that a token trap matching represents the union of those elementary traps that form a smallest marked trap in $N_1 \oplus N_2$. A trap which is fully contained in one of the components and does not touch interface places leads to a trap matching where one X_i is non-empty while both Y_i are empty.

Definition 13 (Formula assigned to N_1 and N_2). *Let N_1 and N_2 be open nets. Then the corresponding formula $\phi(N_1, N_2)$ is built as follows:*

$$\phi(N_1, N_2) = \phi_1 \wedge \phi_2 \wedge \phi_3$$

where, for $i \in \{1, 2\}$, $\Sigma'_i = \{\sigma \mid \sigma \in \Sigma_i, int(\sigma) \subseteq S_{2-i}\}$ *and*

$$\begin{aligned}
\phi_1 &= \bigvee\nolimits_{x \in \Sigma'_1 \cup \Sigma'_2} x \\
\phi_2 &= \bigwedge\nolimits_{x \in \Sigma'_1} (x \Longrightarrow \bigwedge\nolimits_{s \in int(x) \cap S_{i,1}} \bigvee\nolimits_{y \in \Sigma'_2 : s \in int(y)} y) \;\wedge \\
&\quad\; \bigwedge\nolimits_{x \in \Sigma'_2} (x \Longrightarrow \bigwedge\nolimits_{s \in int(x) \cap S_{i,2}} \bigvee\nolimits_{y \in \Sigma'_1 : s \in int(y)} y) \\
\phi_3 &= \bigwedge\nolimits_{[X_1, Y_1, X_2, Y_2] \textit{is internal minimal token trap matching}} \\
&\qquad (\bigvee\nolimits_{N \in X_1} \bigwedge\nolimits_{\Sigma^* \in w_{M1}(N)} \bigvee\nolimits_{\sigma \in \Sigma^*} \neg\sigma \;\vee \\
&\qquad \bigvee\nolimits_{N \in Y_1} \bigwedge\nolimits_{\Sigma^* \in w_{Q_1}(N)} \bigvee\nolimits_{\sigma \in \Sigma^*} \neg\sigma \;\vee \\
&\qquad \bigvee\nolimits_{N \in X_2} \bigwedge\nolimits_{\Sigma^* \in w_{M2}(N)} \bigvee\nolimits_{\sigma \in \Sigma^*} \neg\sigma \;\vee \\
&\qquad \bigvee\nolimits_{N \in Y_2} \bigwedge\nolimits_{\Sigma^* \in w_{Q_2}(N)} \bigvee\nolimits_{\sigma \in \Sigma^*} \neg\sigma)
\end{aligned}$$

Example. For the composition of N_1 and N_2 in Fig. 1, we obtain $\Sigma'_1 = \emptyset$ and $\Sigma'_2 = \{7, 8\}$, so any assignment satisfying ϕ_1 ensures that the second part of ϕ_2 and therefore ϕ_2 overall will be false. Informally this means that all siphons in $N_1 \oplus N_2$ touch the interface of $N_1 \oplus N_2$ so nothing needs to be checked. For obtaining a nontrivial formula, rename e to a and f to b in Fig. 1. In that case, we obtain

- $\phi_1 = 1 \vee \ldots \vee 6 \vee 7 \vee \ldots \vee 12$;
- $\phi_2 = (1 \Longrightarrow (9 \vee 11)) \wedge (2 \Longrightarrow (10 \vee 12)) \wedge (3 \Longrightarrow (9 \vee 11)) \wedge (4 \Longrightarrow (9 \vee 11)) \wedge \ldots \wedge (12 \Longrightarrow (4 \vee 6))$;
- $\phi_3 = true$

ϕ_3 is true as there are no tokens in the system and the empty conjunction is always true. This leads to satisfying assignments. For instance, assigning true to 3 and 9 would satisfy the whole formula. Indeed, the represented siphon $\{a, c, e\}$ does not contain a marked trap.

Assuming a token on c in both components, we would need to include formulas for each internal minimal token trap matching. An example for such a matching is $[\{\{c\}\}, \{\{a, c\}\}, \emptyset, \{\{c, e = a\}\}]$. This matching would contribute the following subformula to $\phi_3 = (\neg 3 \wedge \neg 5) \vee \neg 3 \vee \neg 9$. This subformula states that the trap $\{a, c, e = a\}$ be not included in any siphon represented by a satisfying assignment of the formula. □

Theorem 2. *Let N_1 and N_2 be open nets. $\phi(N_1, N_2)$ is satisfiable if and only if there exists a siphon D of $N_1 \oplus N_2$ such that $D \cap S_1 \cap S_2 \neq \emptyset$ and D does not contain any marked trap.*

Proof. ($\rightarrow$) Let β be a satisfying assignment of $\phi(N_1, N_2)$ and consider the set of places $D_1 \cup D_2$ with $D_1 = \bigcup_{\sigma \in \Sigma_1 : \beta(\sigma) = true} l_1(\sigma)$ and $D_2 = \bigcup_{\sigma \in \Sigma_2 : \beta(\sigma) = true} l_2(\sigma)$. Here, l_i are the mappings used in Def. 12 for N_i, resp. As we composed elementary siphons, D_1 is a siphon of N_1 and D_2 is a siphon of N_2. By ϕ_2, both siphons share the same interface places, $D_1 \cup D_2$ is a siphon of $N_1 \oplus N_2$. ϕ_1 tells us that this siphon is not empty since it contains at least one elementary siphon and elementary siphons cannot be empty. Assume $D_1 \cup D_2$ contains a marked trap. By Lemma 4, it also contains a union of some interface-elementary and one token-elementary trap of N_1 or N_2 or both. A minimal such union defines a token trap matching for which a corresponding subformula is part of ϕ_3. This subformula asserts that for at least one trap participating in the considered union of elementary traps (second level operator), $D_1 \cup D_2$ does not contain sufficiently many elementary siphons to include that elementary trap. In consequence, the whole trap cannot be contained in $D_1 \cup D_2$.

($\leftarrow$) Assume there is a siphon D in $N_1 \oplus N_2$ that contains places in $S_1 \cap S_2$ and does not contain a marked trap. By Lemma 4, D includes a siphon D' that is the union of elementary siphons and which is obviously unmarked as well. Since we only leave out redundant elementary siphons in Def. 11, a siphon D'' can be constructed from the elementary siphons in $\mathcal{M}_{D1}$ and $\mathcal{M}_{D2}$ such that $D'' \cap S_1$ is worse than $D' \cap S_1$ and $D'' \cap S_2$ is worse than $D' \cap S_2$. By Def.7, D'' cannot contain a marked trap either. Consider the assignment β that assigns true to all symbols that represent elementary siphons participating in D''. As D'' has the same (non-empty) set of places in $S_1 \cap S_2$, D'' is not empty. Consequently, D'' includes at least one elementary siphon and thus ϕ_1 must be satisfied. Further, $D'' \cap S_1$ and $D'' \cap S_2$ share the same places in $S_1 \cap S_2$, so ϕ_2 must be satisfied. Finally, since D'' does not contain a marked trap, no union of a subset of the used elementary siphons wraps a marked trap, Thus, each wrap of any marked trap must contain one siphon that is not used to form D''. Consequently, ϕ_3 is satisfied. □

Let us now shift our attention to the open siphons of $N_1 \oplus N_2$. We need to produce the information (according to Def. 11) for $N_1 \oplus N_2$ from the information for N_1 and the one for N_2.

There are two kinds of open siphons and traps in $N_1 \oplus N_2$. First there are those fully contained in one of the components, i.e. disjoint to either S_1 or S_2. They are elementary if and only if they are elementary in their component, and they are wrapped by elements of their own component only. They can be recognised by having no interface places in common with the set of places of the other component. Consequently, information about these siphons and traps can be directly copied from the information provided by the respective component.

Second, there are siphons and traps that spread over both components. Such a siphon (or trap, resp.) is composed of a set of elementary siphons (traps, resp.) of both components. We only need to consider such a siphon if it also contains places in $S_i \cup S_o$ since otherwise it can be decomposed into disjoint siphons (or traps) of the individual components. Thus, the strategy of composing elementary siphons and traps of the components to siphons and traps of $N_1 \oplus N_2$ is to find the smallest sets of individual siphons and traps of N_1 and N_2 that match at $S_1 \cap S_2$. A composite trap is wrapped by a set of siphons if and only if each individual elementary trap is wrapped within its own component and the resulting set of siphons is minimal. All the described information can be computed from the abstracted information that is provided by the components.

Definition 14 (Information for $N_1 \oplus N_2$). *Let, for $i \in \{1,2\}$, $[\Sigma_i, int_i, L_{Q_i}, L_{M_i}, w_{Q_i}, w_{M_i}]$ be the information for N_i. Define the information for $N_1 \oplus N_2$ as $[\Sigma, int, L_Q, L_M, w_Q, w_M]$ with*

- *Σ be the set of minimal siphon matchings between N_1 and N_2;*
- *for each $[Z_1, Z_2] \in \Sigma$, let $int([Z_1, Z_2]) = (\bigcup_{\sigma_1 \in Z_1} int(\sigma_1) \cup \bigcup_{\sigma_2 \in Z_2} int(\sigma_2)) \setminus (S_1 \cap S_2)$;*
- *$L_Q = \{(\bigcup Y_1 \cup \bigcup Y_2) \setminus (S_1 \cap S_2) \mid [Y_1, Y_2]$ is interface trap matching $\}$;*
- *$L_M = \{(\bigcup X_1 \cup \bigcup X_2 \bigcup Y_1 \cup \bigcup Y_2) \setminus (S_1 \cap S_2) \mid [X_1, Y_1, X_2, Y_2]$ is token trap matching $\}$;*
- *$w_Q(X) = \{\{[Z_{11}, Z_{21}], \ldots, [Z_{1k}, Z_{2k}]\} \subseteq \Sigma \mid$ exists minimal interface trap matching $[Y_1, Y_2]$ s.t. $(\bigcup Y_1 \cup \bigcup Y_2) \setminus (S_1 \cap S_2) = X$, and $\forall X' \in Y_1 \exists M \in w_Q(X')\colon M \subseteq \bigcup_{i=1}^k Z_{1i}, \quad \forall X' \in Y_2 \exists M \in w_Q(X')\colon M \subseteq \bigcup_{i=1}^k Z_{2i}\}$.*
- *$w_M(X) = \{\{[Z_{11}, Z_{21}], \ldots, [Z_{1k}, Z_{2k}]\} \subseteq \Sigma \mid$ exists minimal token trap matching $[X_1, Y_1, X_2, Y_2]$ s.t. $(\bigcup Y_1 \cup \bigcup Y_2 \cup \bigcup X_1 \cup \bigcup X_2) \setminus (S_1 \cap S_2) = X$, $\forall X' \in Y_1 \exists M \in w_Q(X')\colon M \subseteq \bigcup_{i=1}^k Z_{1i}, \quad \forall X' \in Y_2 \exists M \in w_Q(X')\colon M \subseteq \bigcup_{i=1}^k Z_{2i}$, $\forall X' \in X_1 \exists M \in w_M(X')\colon M \subseteq \bigcup_{i=1}^k Z_{1i}, \quad \forall X' \in X_2 \exists M \in w_M(X')\colon M \subseteq \bigcup_{i=1}^k Z_{2i}\}$*

Within the values of w_Q and w_M, we silently assume that supersets of other elements are removed.

Example. Let us compose N_1 with N_2 in Fig. 1. We need to consider the following 14 siphon matchings. For convenience, we assign a number to each matching.

$13 = [\{1\}, \emptyset], 14 = [\{2\}, \emptyset], 15 = [\{3\}, \{7\}], 16 = [\{3\}, \{9\}], 17 = [\{3\}, \{10\}], 18 = [\{4\}, \{8\}]$, $19 = [\{4\}, \{11\}]$, $20 = [\{4\}, \{12\}]$, $21 = [\{5\}, \{7\}]$, $22 = [\{5\}, \{9\}]$, $23 = [\{5\}, \{10\}]$, $24 = [\{6\}, \{8\}]$, $25 = [\{6\}, \{11\}]$, $26 = [\{6\}, \{12\}]$. We can represent the interfaces $\{a, e\}$, $\{a, f\}$, $\{b, e\}$, $\{b, f\}$, $\{e\}$, and $\{f\}$ with interface trap matchings, so these six sets form L_Q. L_M is empty as the components do not provide elementary token traps. For computing the wrapping siphons for $\{a, e\}$, we need to consider those trap matchings which generate this interface: $[\{\{a, c\}\}, \{\{c, e\}\}]$ and $[\{\{a, d\}\}, \{\{d, e\}\}]$. $\{a, c\}$ is wrapped by $\{3\}$, $\{c, e\}$ is wrapped by $\{9\}$, $\{a, d\}$ is wrapped by $\{4\}$, and $\{d, e\}$ is wrapped by $\{11\}$. Hence, we need to look into those siphon matchings which contain any of the siphons 3, 4, 9, or 11. Siphon 3 is contained in 15, 16, and 17. Siphon 4 is contained in 18, 19, and 20. In the second component, siphon 9 is contained in 16 and 22. Siphon 11 is contain in 19 and 25. These siphons need to be combined in a minimal way such that either 3 and 9 or 4 and 11 are contained. Hence, we result in $w_Q(\{a, e\}) = \{\{15, 22\}, \{16\}, \{17, 22\}, \{18, 25\}, \{19\}, \{20, 25\}\}$. The remaining values of w_Q can be computed similarly. w_M is empty in the example but the principal approach resembles the one for w_Q. □

Theorem 3. *Let N_1 and N_2 be open nets. Then the information for $N_1 \oplus N_2$ using Def. 14 is equivalent to the information for $N_1 \oplus N_2$ according to Def. 11.*

Proof. It is easy to see that the definition implements the considerations on siphons and traps of the composed system. □

The construction of Def. 14 may introduce redundant information. The conditions of Def. 10 can, however, be evaluated by the information available for $N_1 \oplus N_2$, so information about redundant elementary siphons can be removed after having applied Def. 14.

Example. In the calculation of the previous example, siphons 15, 18, 19, 20, 21, 24, 25, and 26 can be removed through redundancy. Let us verify redundancy for siphon 19. We have to exhibit, for every union U of elementary siphons containing 19, a worse one U' not containing 19. In the example it is quite obvious, that, for each interface, there are two elementary siphons in the composition: one that is obtained using common interface place c, and the other obtained using interface place d. Call these siphons *dual* to each other. The dual to 19 is 16. For a composition of elementary siphons that contains both 16 and 19, let $U' = U \setminus \{19\}$. U' has the same interface as U (since 16 and 19 have the same interface) and it is worse than U as it is composed of less ingredients. Otherwise, let U' be the set of duals to the ones contained in U. 19 is not contained in U' as we assumed that $16 \notin U$. U' has the same interface as U as dual elementary siphons have the same interface. U' includes traps for the same interfaces as traps are symmetric w.r.t. exchanging c and d. After having removed redundant siphons, the remaining information for $N_1 \oplus N_2$ is

- $\Sigma = \{13, 14, 16, 17, 22, 23\}$;
- $int(13) = \{a\}, int(14) = \{b\}, int(16) = \{a, e\}, int(17) = \{a, f\}, int(22) = \{b, e\}, int(23) = \{b, f\}$;

- $L_Q = \{\{a, e\}, \{a, f\}, \{b, e\}, \{b, f\}, \{e\}, \{f\}\}$;
- $L_{M1} = \emptyset$;
- $w_Q(\{a, e\}) = \{\{16\}, \{17, 22\}\}, w_Q(\{a, f\}) = \{\{17\}, \{16, 23\}\}, w_Q(\{b, e\}) = \{\{22\}, \{16, 23\}\}$, $w_Q(\{b, f\}) = \{\{23\}, \{17, 22\}\}, w_Q(\{e\}) = \{\{16\}, \{22\}\}$, $w_Q(\{f\}) = \{\{17\}, \{23\}\}$;
- $w_M = \emptyset$.

This means that the number of elementary siphons as well as the number of interfaces to be considered for traps does not increase during composition. There are only minor differences in w_Q. □

4.4 Discussion

The approach projects information about a component to its interface. The calculations at the interface, e.g. finding matchings or redundancies, appear to be complex, but their complexity depends much more on the size of the interface than on the size of the net behind the interface. In the running example, we may compose longer and longer chains or rings of components such as in Fig. 1. As each resulting component has information similar to the one for single components, the overall complexity grows linearly with the number of components to be composed. In comparison, the resulting net has an exponentially growing number of minimal siphons (since every circle where either the upper or the lower place is taken forms a minimal siphon). We conclude that the divide-and-conquer approach is beneficial at least in those cases where a decomposition exists such that intermediate interfaces during re-composition remain small. How to obtain such a decomposition in general remains to be seen. Since we may switch to the original algorithm for computing elementary siphons and traps at any stage of decomposition, it is possible to apply the divide-and-conquer strategy whenever the size of the interface between components is significantly smaller than the inner structure of a component. Consequently, we may judge that the proposed strategy is rather valuable even though we cannot provide experimental evidence at this time.

5 Conclusion

We proposed two new approaches to deciding the siphon trap property and thus to getting information about important properties like liveness or deadlock freedom. One approach is a straight transformation to the SAT problem for which we inherit the sophistication of existing SAT solvers. The second approach uses the well known divide-and-conquer strategy. It is based on a known decomposition into open nets which we refined such that we may arrive at arbitrarily small components, and on a projection of information about siphons and traps to the interface of a component. This way, complexity expresses itself more in terms of the size of interfaces than of the size of the component as such which makes the procedure applicable at least for certain classes of nets with somewhat sparse connectivity.

For the first approach, the main remaining issue is to get to smaller formulae. In particular, we frequently copy a certain subformula. There may be structural considerations for reducing the number of required copies for certain net classes. In the divide-and-conquer approach, there are still some nondeterministic choices. We need to provide a prototype implementation including heuristics for these choices for further underpinning its usefulness. In addition, we would like to have a reasonable criterion for verifying Def. 10.

References

1. Barkaoui, K., Minoux, M.: A polynomial-time graph algorithm to decide liveness of some basic classes of bounded Petri nets. In: Jensen, K. (ed.) ICATPN 1992. LNCS, vol. 616, pp. 62–75. Springer, Heidelberg (1992)
2. Desel, J., Esparza, J.: Free Choice Petri nets. Cambridge Tracts in Theoretical Computer Science, vol. 40. Cambridge University Press, Cambridge (1995)
3. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
4. Esparza, J., Nielsen, M.: Decidability issues for Petri nets. Petri Nets Newsletter 52, 245–262 (1994)
5. Hack, M.H.T.: Analysis of Production Schemata by Petri Nets. Master's thesis, MIT, Dept. Electrical Engineering, Cambridge, Mass (1972)
6. INA. Integrated Net Analyzer (2003), http://www2.informatik.hu-berlin.de/~starke/ina.html
7. Karatkevich, A.: Analysis by solving logical equations – calculation of siphons and traps. In: Dynamic Analysis of Petri Net-based Discrete Systems. LNCIS, vol. 356, pp. 87–93. Springer, Heidelberg (2007)
8. Mennicke, S., Oanea, O., Wolf, K.: Decomposition into open nets. In: AWPN 2009. CEUR Workshop Proceedings, vol. 501, pp. 29–34. CEUR-WS.org (2009)
9. MiniSat. Minimalistic, open-source SAT solver (2007), http://www.minisat.se
10. Minoux, M., Barkaoui, K.: Polynomial algorithms for proving or disproving Commoner's property in Petri nets. In: Proceedings 9th Workshop on Theory and Applications of Petri Nets, vol. 1, pp. 113–125 (1988)
11. Zaitsev, D.A.: Decomposition of Petri nets. Cybernetics and Systems Analysis (5), 131–140 (2004)