# An Upper Bound for the Reachability Problem of Safe, Elementary Hornets

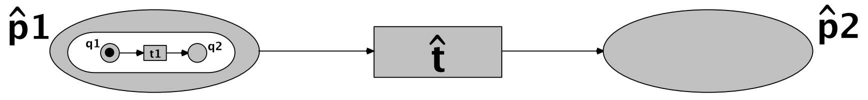
Michael Köhler-Bußmeier and Frank Heitmann

University of Applied Science Hamburg and University of Hamburg
michael.koehler-bussmeier@haw-hamburg.de

**Abstract.** In this paper we study the complexity of the reachability problem Hornets, an algebraic extension of object nets. Here we consider the restricted class of safe, elementary Hornets.
In previous work we established the lower bound, i.e. reachability requires at least exponential space. In another work we have shown we can simulate elementary Hornets with elemntary object nets Eos, where reachability is known to be PSpace-complete. Since this simulation leads to a double exponetial increase in the size of the simulating Eos, we obtain that for Hornets the reachability problem is solvable in double exponetial space.
In this contribution we show that the simulation is rather bad, since we show that exponential space is sufficient, i.e. upper and lower bound coincide.
**Keywords:** Hornets, nets-within-nets, object nets, reachability, safeness

## 1 Hornets: Higher-Order Object Nets

In this paper we study the algebraic extension of object nets, called Hornets. Hornets are a generalisation of object nets [1, 2], which follow the *nets-within-nets* paradigm as proposed by Valk [3]. With object nets we study Petri nets where the tokens are nets again, i.e. we have a nested marking. Events are also nested. We have three different kinds of events – as illustrated by the following example:



1. System-autonomous: The system net transition $\widehat{t}$ fires autonomously, which moves the net-token from $\widehat{p}_1$ to $\widehat{p}_2$ without changing its marking.
2. Object-autonomous: The object net fires transition $t_1$ "moving" the black token from $q_1$ to $q_2$. The object net remains at its location $\widehat{p}_1$.
3. Synchronisation: Whenever we add matching synchronisation inscriptions at the system net transition $\widehat{t}$ and the object net transition $t_1$, then both must fire synchronously: The object net is moved to $\widehat{p}_2$ and the black token moves from $q_1$ to $q_2$ inside. Whenever synchronisation is specified, autonomous actions are forbidden.

For HORNETS we extend object-nets with algebraic concepts that allow to modify the structure of the net-tokens as a result of a firing transition. This is a generalisation of the approach of algebraic nets [4], where algebraic data types replace the anonymous black tokens.

*Example 1.* We consider a HORNET with two workflow nets $N_1$ and $N_2$ as tokens – cf. Figure 1. To model a run-time adaption, we combine $N_1$ and $N_2$ resulting in the net $N_3 = (N_1 \| N_2)$. This modification is modelled by transition $t$ of the HORNETS in Fig. 1. In a binding $\alpha$ with $x \mapsto N_1$ and $y \mapsto N_2$ the transition $t$ is enabled. Assume that $(x \| y)$ evaluates to $N_3$ for $\alpha$. If $t$ fires it removes the two net-tokens from $p$ and $q$ and generates one new net-token on place $r$. The net-token on $r$ has the structure of $N_3$ and its marking is obtained as a transfer from the token on $v$ in $N_1$ and the token on $s$ in $N_2$ into $N_3$. This transfer is possible since all the places of $N_1$ and $N_2$ are also places in $N_3$ and tokens can be transferred in the obvious way.

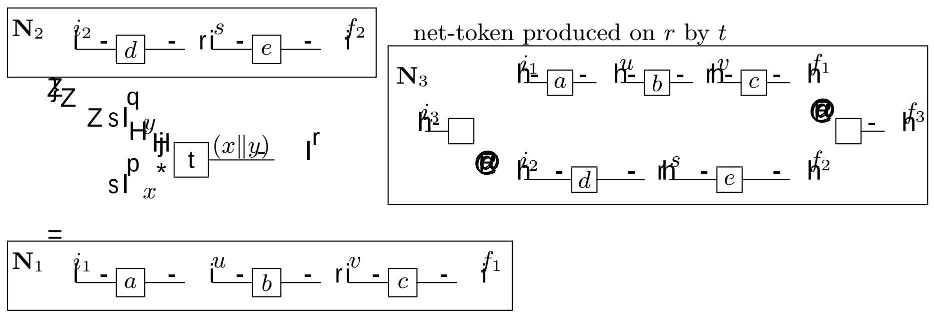


Fig. 1. Modification of the net-token's structure

The use of algebraic operations in HORNETS relates them to *algebraic higher-order (AHO) systems* [5], which are restricted to two-levelled systems but have a greater flexibility for the operations on net-tokens, since each net transformation is allowed. There is also a relationship to Nested Nets [6], which are used for adaptive systems.

It is not hard to prove that the general HORNET formalism is Turing-complete. In [7] we have proven that there are several possibilities to simulate counter programs: One could use the nesting to encode counters. Another possibility is to encode counters in the algebraic structure of the net operators.

In this paper we like to study the *complexity* that arises due the algebraic structure. Here, we restrict HORNETS to guarantee that the system has a finite state space. First, we allow at most one token on each place, which results in the class of *safe* HORNETS. However this restriction does not guarantee finite state spaces, since we have the nesting depth as a second source of undecidability [2]. Second, we restrict the universe of object nets to finite sets. Finally, we restrict the nesting depth and introduce the class of *elementary* HORNETS, which have a two-levelled nesting structure. This is done in analogy to the class of *elemtary* *object net systems (*Eos*)* [1], which are the two-level specialisation of general object nets [1, 2].

If we rule out these sources of complexity the main origin of complexity is the use of algebraic transformations, which are still allowed for safe, elementary Hornets. As a result we obtain the class of safe, elementary Hornets – in analogy to the class of *safe* Eos [8].

We have shown in [8, 9] that most problems for *safe* Eos are PSpace-complete. More precisely: All problems that are expressible in LTL or CTL, which includes reachability and liveness, are PSpace-complete. This means that with respect to these problems *safe* Eos are no more complex than p/t nets.

In a previous publication [10] we have shown that *safe, elementary* Hornets are beyond PSpace. We have shown a lower bound, i.e. that "the reachability problem requires exponential space" for safe, elementary Hornets – similarily to well known result of for *bounded* p/t nets [11]. In [12] we have shown that each elementary Hornet $EH$ can be simulated by an Eos $OS(G(EH))$. Therefore, we can reduce the reachability problem for *safe, elementary* Hornets to the one for *safe* Eos.

Unfortunaltely, the construction of the simulating Eos $OS(G(EH))$ as given in Theorem 3 of [12] leads to an Eos that is double-exponentially larger than the Hornet $EH$, which establishes the fact that the reachability problem for safe, elementary Hornets requires at most double-exponential space.

In this paper we give establish an upper bound in a more direct fashion. We will give an decision procedure which needs at most exponential space, which shows that lower and upper bound coincide.

The paper has the following structure: Section 2 defines Elementary Hornets. Since the reachability problem is known to be undecidable even for Eos, we restrict elementary Hornets to safe ones. Safe, elementary Hornets have finite state spaces. In Section 3 we present a decision procedure for reachability that operates within exponential space. Since we know that exponential space is also a lower bound for the problem, the algorithm is optimal. The work ends with a conclusion.

## 2 Definition of Elementary Hornets

A multiset $\mathbf{m}$ on the set $D$ is a mapping $\mathbf{m} : D \to \mathsf{N}$. Multisets can also be represented as a formal sum in the form $\mathbf{m} = \sum_{i=1}^{n} x_i$, where $x_i \in D$.

Multiset addition is defined component-wise: $(\mathbf{m}_1 + \mathbf{m}_2)(d) := \mathbf{m}_1(d) + \mathbf{m}_2(d)$. The empty multiset $\mathbf{0}$ is defined as $\mathbf{0}(d) = 0$ for all $d \in D$. Multiset-difference $\mathbf{m}_1 - \mathbf{m}_2$ is defined by $(\mathbf{m}_1 - \mathbf{m}_2)(d) := \max(\mathbf{m}_1(d) - \mathbf{m}_2(d), 0)$.

The cardinality of a multiset is $|\mathbf{m}| := \sum_{d \in D} \mathbf{m}(d)$. A multiset $\mathbf{m}$ is finite if $|\mathbf{m}| < \infty$. The set of all finite multisets over the set $D$ is denoted $MS(D)$.

Multiset notations are used for sets as well. The meaning will be apparent from its use.

Any mapping $f : D \to D'$ extends to a multiset-homomorphism $f^\sharp : MS(D) \to MS(D')$ by $f^\sharp \left(\sum_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} f(x_i)$.

A *p/t net* $N$ is a tuple $N = (P, T, \mathbf{pre}, \mathbf{post})$, such that $P$ is a set of places, $T$ is a set of transitions, with $P \cap T = \emptyset$, and $\mathbf{pre}, \mathbf{post} : T \to MS(P)$ are the pre- and post-condition functions. A marking of $N$ is a multiset of places: $\mathbf{m} \in MS(P)$. We denote the enabling of $t$ in marking $\mathbf{m}$ by $\mathbf{m} \xrightarrow{t}$. Firing of $t$ is denoted by $\mathbf{m} \xrightarrow{t} \mathbf{m}'$.

## 2.1 Elementary Hornets (eHornets)

*Net-Algebras* We define the algebraic structure of object nets. For a general introduction of algebraic specifications cf. [13].

Let $K$ be a set of net-types (kinds). A (many-sorted) *specification* $(\Sigma, X, E)$ consists of a signature $\Sigma$, a family of variables $X = (X_k)_{k\in K}$, and a family of axioms $E = (E_k)_{k\in K}$.

A signature is a disjoint family $\Sigma = (\Sigma_{k_1\cdots k_n,k})_{k_1,\cdots,k_n,k\in K}$ of operators. The set of terms of type $k$ over a signature $\Sigma$ and variables $X$ is denoted $\mathsf{T}^k_\Sigma(X)$.

We use (many-sorted) predicate logic, where the terms are generated by a signature $\Sigma$ and formulae are defined by a family of predicates $\Psi = (\Psi_n)_{n\in\mathbb{N}}$. The set of formulae is denoted $PL_\Gamma$, where $\Gamma = (\Sigma, X, E, \Psi)$ is the *logic structure*.

Let $\Sigma$ be a signature over $K$. A *net-algebra* assigns to each type $k \in K$ a set $\mathcal{U}_k$ of object nets. Each object $N \in \mathcal{U}_k, k \in K$ net is a p/t net $N = (P_N, T_N, \mathbf{pre}_N, \mathbf{post}_N)$. We identify $\mathcal{U}$ with $\bigcup_{k\in K} \mathcal{U}_k$ in the following. We assume the family $\mathcal{U} = (\mathcal{U}_k)_{k\in K}$ to be disjoint.

The nodes of the object nets in $\mathcal{U}_k$ are not disjoint, since the firing rule allows to transfer tokens between net tokens within the same set $\mathcal{U}_k$. Such a transfer is possible, if we assume that all nets $N \in \mathcal{U}_k$ have the same set of places $P_k$. $P_k$ is the place universe for all object nets of kind $k$. In the example of Fig. 1 the object nets $N_1$, $N_2$, and $N_3$ must belong to the same type since otherwise it would be impossible to transfer the markings in $N_1$ and $N_2$ to the generated $N_3$.

In general, $P_k$ is not finite. Since we like each object net to be finite in some sense, we require that the transitions $T_N$ of each $N \in \mathcal{U}_k$ use only a finite subset of $P_k$, i.e. $\forall N \in \mathcal{U} : |{}^\bullet T_N \cup T_N{}^\bullet| < \infty$.

The family of object nets $\mathcal{U}$ is the universe of the algebra. A net-algebra $(\mathcal{U}, \mathcal{I})$ assigns to each constant $\sigma \in \Sigma_{\lambda,k}$ an object net $\sigma^\mathcal{I} \in \mathcal{U}_k$ and to each operator $\sigma \in \Sigma_{k_1\cdots k_n,k}$ with $n > 0$ a mapping $\sigma^\mathcal{I} : (\mathcal{U}_{k_1} \times \cdots \times \mathcal{U}_{k_n}) \to \mathcal{U}_k$.

A net-algebra is called *finite* if $P_k$ is a finite set for each $k \in K$.

Since all nets $N \in \mathcal{U}_k$ have the same set of places $P_k$, which is finite for eHornets, there is an upper bound for the cardinality of $\mathcal{U}_k$.

**Proposition 1 (Lemma 2.1 in [10]).** *For each $k \in K$ we have $|\mathcal{U}_k| \leq 2^{\left(2^{4|P_k|}\right)}$.*

A variable assignment $\alpha = (\alpha_k : X_k \to \mathcal{U}_k)_{k\in K}$ maps each variable onto an element of the algebra. For a variable assignment $\alpha$ the evaluation of a term $t \in \mathsf{T}^k_\Sigma(X)$ is uniquely defined and will be denoted as $\alpha(t)$.

A net-algebra, such that all axioms of $(\Sigma, X, E)$ are valid, is called *net-theory*.

*Nested Markings* A marking of an eHORNET assigns to each system net place one or many net-tokens. The places of the system net are typed by the function $k : \widehat{P} \to K$, meaning that a place $\widehat{p}$ contains net-tokens of kind $k(\widehat{p})$. Since the net-tokens are instances of object nets, a *marking* is a *nested* multiset of the form:

$$\mu = \sum_{i=1}^{n} \widehat{p}_i[N_i, M_i] \quad \text{where} \quad \widehat{p}_i \in \widehat{P}, N_i \in \mathcal{U}_{\mathsf{k}(\widehat{p}_i)}, M_i \in MS(P_{N_i}), n \in \mathsf{N}$$

Each addend $\widehat{p}_i[N_i, M_i]$ denotes a net-token on the place $\widehat{p}_i$ that has the structure of the object net $N_i$ and the marking $M_i \in MS(P_{N_i})$. The set of all nested multisets is denoted as $\mathcal{M}_H$. We define the partial order $\sqsubseteq$ on nested multisets by setting $\mu_1 \sqsubseteq \mu_2$ iff $\exists \mu : \mu_2 = \mu_1 + \mu$.

The projection $\Pi_N^{1,H}(\mu)$ is the multiset of all system-net places that contain the object-net $N$:[1]

$$\Pi_N^{1,H}\left(\sum\nolimits_{i=1}^{n} \widehat{p}_i[N_i, M_i]\right) := \sum\nolimits_{i=1}^{n} \mathbf{1}_N(N_i) \cdot \widehat{p}_i \tag{1}$$

where the indicator function $\mathbf{1}_N$ is defined as: $\mathbf{1}_N(N_i) = 1$ iff $N_i = N$.

Analogously, the projection $\Pi_N^{2,H}(\mu)$ is the multiset of all net-tokens' markings (that belong to the object-net $N$):

$$\Pi_N^{2,H}\left(\sum\nolimits_{i=1}^{n} \widehat{p}_i[N_i, M_i]\right) := \sum\nolimits_{i=1}^{n} \mathbf{1}_k(N_i) \cdot M_i \tag{2}$$

The projection $\Pi_k^{2,H}(\mu)$ is the sum of all net-tokens' markings belonging to the same type $k \in K$:

$$\Pi_k^{2,H}(\mu) := \sum\nolimits_{N \in \mathcal{U}_k} \Pi_N^{2,H}(\mu) \tag{3}$$

*Synchronisation* The transitions in an HORNET are labelled with synchronisation inscriptions. We assume a fixed set of channels $C = (C_k)_{k \in K}$.

- The function family $\widehat{l}_\alpha = (\widehat{l}_\alpha^k)_{k \in K}$ defines the synchronisation constraints. Each transition of the system net is labelled with a multiset $\widehat{l}^k(\widehat{t}) = (e_1, c_1) + \cdots + (e_n, c_n)$, where the expression $e_i \in \mathsf{T}_\Sigma^k(X)$ describes the called object net and $c_i \in C_k$ is a channel. The intention is that $\widehat{t}$ fires synchronously with a multiset of object net transitions with the same multiset of labels. Each variable assignment $\alpha$ generates the function $\widehat{l}_\alpha^k(\widehat{t})$ defined as:

$$\widehat{l}_\alpha^k(\widehat{t})(N) := \sum\nolimits_{\substack{1 \le i \le n \\ \alpha(e_i) = N}} c_i \quad \text{for} \quad \widehat{l}^k(\widehat{t}) = \sum\nolimits_{1 \le i \le n} (e_i, c_i) \tag{4}$$

  Each function $\widehat{l}_\alpha^k(\widehat{t})$ assigns to each object net $N$ a multiset of channels.
- For each $N \in \mathcal{U}_k$ the function $l_N$ assigns to each transition $t \in T_N$ either a channel $c \in C_k$ or $\bot_k$, whenever $t$ fires without synchronisation, i.e. autonously.

[1] The superscript $H$ indicates that the function is used for HORNETS.

*System Net* Assume we have a fixed logic $\Gamma = (\Sigma, X, E, \Psi)$ and a net-theory $(\mathcal{U}, \mathcal{I})$. An *elementary higher-order object net* (eHornet) is composed of a system net $\widehat{N}$ and the set of object nets $\mathcal{U}$. W.l.o.g. we assume $\widehat{N} \notin \mathcal{U}$. To guarantee finite algebras for eHornets, we require that the net-theory $(\mathcal{U}, \mathcal{I})$ is finite, i.e. each place universe $P_k$ is finite.

The system net is a net $\widehat{N} = (\widehat{P}, \widehat{T}, \mathbf{pre}, \mathbf{post}, \widehat{G})$, where each arc is labelled with a multiset of terms: $\mathbf{pre}, \mathbf{post} : \widehat{T} \to (\widehat{P} \to MS(\mathsf{T}_\Sigma(X)))$. Each transition is labelled by a guard predicate $\widehat{G} : \widehat{T} \to PL_\Gamma$. The places of the system net are typed by the function $k : \widehat{P} \to K$. As a typing constraint we have that each arc inscription has to be a multiset of terms that are all of the kind that is assigned to the arc's place:

$$\mathbf{pre}(\widehat{t})(\widehat{p}), \quad \mathbf{post}(\widehat{t})(\widehat{p}) \quad \in \quad MS(\mathsf{T}_\Sigma^{k(\widehat{p})}(X)) \tag{5}$$

For each variable binding $\alpha$ we obtain the evaluated functions $\mathbf{pre}_\alpha, \mathbf{post}_\alpha : \widehat{T} \to (\widehat{P} \to MS(\mathcal{U}))$ in the obvious way.

**Definition 1 (Elementary Hornet, eHornet).** *Assume a fixed many-sorted predicate logic $\Gamma = (\Sigma, X, E, \Psi)$.*

*An elementary* Hornet *is a tuple $EH = (\widehat{N}, \mathcal{U}, \mathcal{I}, k, l, \mu_0)$ such that:*

1. *$\widehat{N}$ is an algebraic net, called the* system net.
2. *$(\mathcal{U}, \mathcal{I})$ is a finite net-theory for the logic $\Gamma$.*
3. *$k : \widehat{P} \to K$ is the typing of the system net places.*
4. *$l = (\widehat{l}, l_N)_{N \in \mathcal{U}}$ is the labelling.*
5. *$\mu_0 \in \mathcal{M}_H$ is the initial marking.*

## 2.2 Events and Firing Rule

The synchronisation labelling generates the set of system events $\Theta$:

The labelling introduces three cases of events:

1. Synchronised firing: There is at least one object net that has to be synchronised, i.e. there is a $N$ such that $\widehat{l}(\widehat{t})(N)$ is not empty.
   Such an event is a pair $\theta = \widehat{t}^\alpha[\vartheta]$, where $\widehat{t}$ is a system net transition, $\alpha$ is a variable binding, and $\vartheta$ is a function that maps each object net to a multiset of its transitions, i.e. $\vartheta(N) \in MS(T_N)$. It is required that $\widehat{t}$ and $\vartheta(N)$ have matching multisets of labels, i.e. $\widehat{l}(\widehat{t})(N) = l_N^\sharp(\vartheta(N))$ for all $N \in \mathcal{U}$. (Remeber that $l_N^\sharp$ denotes the multiset extension of $l_N$.)
   The intended meaning is that $\widehat{t}$ fires synchronously with all the object net transitions $\vartheta(N), N \in \mathcal{U}$.
2. System-autonomous firing: The transition $\widehat{t}$ of the system net fires autonomously, whenever $\widehat{l}(\widehat{t})$ is the empty multiset $\mathbf{0}$.
   We consider system-autonomous firing as a special case of synchronised firing generated by the function $\vartheta_{id}$, defined as $\vartheta_{id}(N) = \mathbf{0}$ for all $N \in \mathcal{U}$.

3. Object-autonomous firing: An object net transition $t$ in $N$ fires autonomously, whenever $l_N(t) = \bot_k$.
   Object-autonomous events are denoted as $id_{\widehat{p},N}[\vartheta_t]$, where $\vartheta_t(N') = \{t\}$ if $N = N'$ and $\mathbf{0}$ otherwise. The meaning is that in object net $N$ fires $t$ autonomously within the place $\widehat{p}$.
   For the sake of uniformity we define for an arbitrary binding $\alpha$:
$$\mathbf{pre}_\alpha(id_{\widehat{p},N})(\widehat{p}')(N') = \mathbf{post}_\alpha(id_{\widehat{p},N})(\widehat{p}')(N') = \begin{cases} 1 & \text{if } \widehat{p}' = \widehat{p} \wedge N' = N \\ 0 & \text{otherwise.} \end{cases}$$

The set of all events generated by the labelling $l$ is $\Theta_l := \Theta_1 \cup \Theta_2$, where $\Theta_1$ contains synchronous events (including system-autonomous events as a special case) and $\Theta_2$ contains the object-autonomous events:

$$\begin{aligned} \Theta_1 &:= \left\{ \widehat{\tau}^\alpha[\vartheta] \quad \mid \forall N \in \mathcal{U} : \widehat{l}_\alpha(\widehat{t})(N) = l^\sharp_N(\vartheta(N)) \right\} \\ \Theta_2 &:= \left\{ id_{\widehat{p},N}[\vartheta_t] \mid \widehat{p} \in \widehat{P}, N \in \mathcal{U}_{k(\widehat{p})}, t \in T_N \right\} \end{aligned} \tag{6}$$

*Firing Rule* A system event $\theta = \widehat{\tau}^\alpha[\vartheta]$ removes net-tokens together with their individual internal markings. Firing the event replaces a nested multiset $\lambda \in \mathcal{M}_H$ that is part of the current marking $\mu$, i.e. $\lambda \sqsubseteq \mu$, by the nested multiset $\rho$. The enabling condition is expressed by the *enabling predicate* $\phi_{EH}$ (or just $\phi$ whenever $EH$ is clear from the context):

$$\begin{aligned} \phi_{EH}(\widehat{\tau}^\alpha[\vartheta], \lambda, \rho) \iff \forall k \in K :& \\ & \forall \widehat{p} \in k^{-1}(k) : \forall N \in \mathcal{U}_k : \Pi^{1,H}_N(\lambda)(\widehat{p}) = \mathbf{pre}_\alpha(\widehat{\tau})(\widehat{p})(N) \wedge \\ & \forall \widehat{p} \in k^{-1}(k) : \forall N \in \mathcal{U}_k : \Pi^{1,H}_N(\rho)(\widehat{p}) = \mathbf{post}_\alpha(\widehat{\tau})(\widehat{p})(N) \wedge \\ & \Pi^{2,H}_k(\lambda) \geq \textstyle\sum_{N \in \mathcal{U}_k} \mathbf{pre}^\sharp_N(\vartheta(N)) \wedge \\ & \Pi^{2,H}_k(\rho) = \Pi^{2,H}_k(\lambda) + \textstyle\sum_{N \in \mathcal{U}_k} \mathbf{post}^\sharp_N(\vartheta(N)) - \mathbf{pre}^\sharp_N(\vartheta(N)) \end{aligned} \tag{7}$$

The predicate $\phi_{EH}$ has the following meaning: Conjunct (1) states that the removed sub-marking $\lambda$ contains on $\widehat{p}$ the right number of net-tokens, that are removed by $\widehat{\tau}$. Conjunct (2) states that generated sub-marking $\rho$ contains on $\widehat{p}$ the right number of net-tokens, that are generated by $\widehat{\tau}$. Conjunct (3) states that the sub-marking $\lambda$ enables all synchronised transitions $\vartheta(N)$ in the object $N$. Conjunct (4) states that the marking of each object net $N$ is changed according to the firing of the synchronised transitions $\vartheta(N)$.

Note, that conjunct (1) and (2) assures that only net-tokens relevant for the firing are included in $\lambda$ and $\rho$. Conditions (3) and (4) allow for additonal tokens in the net-tokens.

For system-autonomous events $\widehat{t}^\alpha[\vartheta_{id}]$ the enabling predicate $\phi_{EH}$ can be simplified further: Conjunct (3) is always true since $\mathbf{pre}_N(\vartheta_{id}(N)) = \mathbf{0}$. Conjunct (4) simplifies to $\Pi^{2,H}_k(\rho) = \Pi^{2,H}_k(\lambda)$, which means that no token of the object nets get lost when a system-autonomous events fires.

Analogously, for an object-autonomous event $\widehat{\tau}[\vartheta_t]$ we have an idle-transition $\widehat{\tau} = id_{\widehat{p},N}$ and $\vartheta = \vartheta_t$ for some $t$. Conjunct (1) and (2) simplify to $\Pi^{1,H}_{N'}(\lambda) =$

$\widehat{p} = \mathit{\Pi}^{1,H}_{N'}(\rho)$ for $N' = N$ and to $\mathit{\Pi}^{1,H}_{N'}(\lambda) = \mathbf{0} = \mathit{\Pi}^{1,H}_{N'}(\rho)$ otherwise. This means that $\lambda = \widehat{p}[M]$, $M$ enables $t$, and $\rho = \widehat{p}[M - \mathbf{pre}_N(\widehat{t}) + \mathbf{post}_N(\widehat{t})]$.

**Definition 2 (Firing Rule).** *Let EH be an* eHornet *and $\mu, \mu' \in \mathcal{M}_H$ markings.*

- *The event $\widehat{\tau}^\alpha[\vartheta]$ is enabled in $\mu$ for the mode $(\lambda, \rho) \in \mathcal{M}_H^2$ iff $\lambda \sqsubseteq \mu \wedge \phi_{EH}(\widehat{\tau}[\vartheta], \lambda, \rho)$ holds and the guard $\widehat{G}(\widehat{t})$ holds, i.e. $E \models^\alpha_{\mathcal{I}} \widehat{G}(\widehat{\tau})$.*
- *An event $\widehat{\tau}^\alpha[\vartheta]$ that is enabled in $\mu$ can fire – denoted $\mu \xrightarrow[EH]{\widehat{\tau}^\alpha[\vartheta](\lambda,\rho)} \mu'$.*
- *The resulting successor marking is defined as $\mu' = \mu - \lambda + \rho$.*

Note, that the firing rule has no a-priori decision how to distribute the marking on the generated net-tokens. Therefore we need the mode $(\lambda, \rho)$ to formulate the firing of $\widehat{\tau}^\alpha[\vartheta]$ in a functional way.

## 3 Reachability for Hornets

In the following we study the complexity of the reachability problem for safe eHornets.

A Hornet is safe iff each place $\widehat{p}$ in the system net carries at most one token and each net-token carries at most one token on each place $p$ in all reachable markings $\mu$. Since we are interested in safe Hornets in the following, we do not use arc weights greater than 1.

**Proposition 2 (Lemma 3.1 [10]).** *A safe* eHornet *has a finite reachability set. There are at most $\left(1 + 2^{\left(2^{4m}\right)} \cdot 2^m\right)^{|\widehat{P}|}$ different markings, where $m$ is the maximum of all $|P_k|$.*

The huge number of object nets in the set $\mathcal{U}_k$ of object nets and reachable markings (cf. Prop. 2) leads to the following result.

**Proposition 3 (Theorem 5.1 in [10]).** *The reachability problem for safe* eHornets *requires at least exponential space.*

The proof we have given in [10] is similar to the heart of Lipton's famous result. In [10] we simulate a counter program $C$ by a safe eHornet $EH$ of size $O(\mathit{poly}(|C|))$. Thus we know that a question about a counter program $C$ translates into a question about $EH$. When $C$ needs $2^{|A|}$ space for the counters, we have that $EH$ needs $2^{\sqrt[n]{|A|}}$ space for the corresponding question, where $n$ is the order of the polynom $\mathit{poly}(|C|)$.

In [12] we have given a reduction of the reachability problem for safe eHornets to that of safe Eos. We have shown, that for a safe eHornet $EH$ the simulating Eos $OS(EH)$ is safe, too. Since we know that the reachability problem for safe Eos is PSpace-complete [8], we can use the simulating Eos to decide the reachability problem for $EH$. We have shown that the simulating

Eos $OS(G(EH))$ increases double-exponentially in size. Let $m$ be the maximum of all $|P_k|$. Then, we obtain a space complexity of:

$$POLY\left(2^{O\left(2^{4m}\right)}\right) = 2^{\mathsf{POLY}\cdot O\left(2^{4m}\right)} = 2^{2^{O(m)}}$$

**Proposition 4 (Theorem 3 in [12]).** *The reachability problem $\mathsf{Reach_{seH}}$ for safe* eHornets *is in $DSpace\left(2^{2^{O(m)}}\right)$.*

This implies quite a big gap between the lower bound (exponential space) and the upper bound resulting from the simulation (double-exponential space).[2] Fortunately, simulation is not the only way to obtain an upper complexity bound. In the following we give a more direct approach with a better complexity. As a first step we give a non-deterministic algorithm to decide reachability of $\mu_1$.

In the following, we assume that all evaluations of terms and predicates need at most $2^{O(m)}$ bits.

**Lemma 1.** *There exists a non-determistic algorithm that decides the reachability problem $\mathsf{Reach_{seH}}$ for safe* eHornets *within exponential space: $\mathsf{Reach_{seH}} \in NSpace(2^{O(m)})$, where $m$ is the maximum of all $|P_k|$.*

*Proof.* The main idea of the following algorithm is to guess a firing sequence $\mu_0 \xrightarrow{\theta_1} \mu_1 \xrightarrow{\theta_2} \cdots \xrightarrow{\theta_n} \mu_n = \mu^*$, where $\mu^*$ is the marking to be tested for reachability.

By Prop. 2 we know we have at most $2^{O(2^m)}$ different markings in the safe Hornet. Therefore, we can safely cut off the computation after $2^{2^{O(m)}}$ steps – whenever $\mu^*$ is reachable it is reachable by a firing sequence without loops.

```
function $reach(EH$: eHornet, $\mu^*$: Marking): boolean is
  $c := 2^{2^{O(m)}}$
  $\mu := \mu_0$
  found := $(\mu = \mu^*)$
  while $\neg$found $\wedge\ c > 0$ do
      choose non-deterministically an event $\widehat{t}^{\alpha}[\vartheta](\lambda, \rho)$
      if $\mu \xrightarrow[EH]{\widehat{t}^{\alpha}(\lambda,\rho)}$ then
          $\mu := \mu - \lambda + \rho$     // the resulting successor marking
          $c := c - 1$
          found := $(\mu = \mu^*)$
      else
          return false
  end while
  return found
```

[2] This was to expect, since we have a long 'chain' of "recycled" techniques: the simulating Eos $OS(G(\mathsf{EH}))$ and the PSpace algorithm for Eos.

In each step $\mu_i \xrightarrow{\theta_i} \mu_{i+1}$ we choose non-deterministically some event $\theta_i$.

How much space is needed to store an event? Note, that each common set of places $P_k$ is finite for eHornets. Each possible transition $t$ chooses a subset of $P_k$ for the preset ${}^\bullet t$ and another subset for the postset $t^\bullet$. We identify $t$ with the pair $({}^\bullet t, t^\bullet)$. A transition is an element from $T_k := \mathcal{P}(P_k) \times \mathcal{P}(P_k)$. We have $|T_k| = 2^{|P_k|} \cdot 2^{|P_k|} = (2^{|P_k|})^2 = 2^{2|P_k|}$ different transitions.

To each transition $t \in T_N$, $N \in \mathcal{U}_k$ the partial function $l_N$ assigns either a channel $c \in C_k$ or $\bot_k$.

The set of labelled transitions is $LT_k := T_k \times (C_k \cup \{\bot_k\}))$ and we have $|LT_k| = |T_k \times (C_k \cup \{\bot_k\})| = 2^{2|P_k|} \cdot (|C_k| + 1)$ different labelled transitions.

We cannot use more channels than we have transitions in the object net, i.e. we could use at most $|T_k| \leq 2^{2|P_k|}$ different channels from $C_k \cup \{\bot_k\}$. Thus, we have:

$$|LT_k| = 2^{2|P_k|} \cdot (|C_k| + 1) \leq 2^{2|P_k|} \cdot 2^{2|P_k|} \leq 2^{4|P_k|}$$

Now we can quantify the amount of space needed for the event $\theta$:

1. We select some system net transition $\widehat{t}$.
2. We guess a variable binding $\alpha$. For each variable in $var(\widehat{t})$ of kind $k$ we select some object net from the universe $\mathcal{U}_k$. We have $\mathcal{U}_k|^{|var(\widehat{t})|} \leq 2^{\left(|var(\widehat{t})| \cdot 2^{4m}\right)}$ candidates for the obejct net. Each choice needs $\log\left(|var(\widehat{t})| \cdot 2^{4m}\right) \in 2^{O(m)}$ bits of space. (Compared to the other aspects, the variable binding $\alpha$ uses a major part of the space needed.)
3. For each kind $k$ we have the multiset of channels $\widehat{l}^k_\alpha(\widehat{t})$ to synchronise with. Whenever $\widehat{l}^k_\alpha(\widehat{t})(N)$ is not the empty multiset $\mathbf{0}$, then we have to select a multiset of transitions from $T_N$) (with corresponding channel inscriptions). The number of channels used is $|\widehat{l}^k_\alpha(\widehat{t})|$. For each of these $|\widehat{l}^k_\alpha(\widehat{t})|$ channels we have non-deteministically select one labelled object net transition from the set $LT_k$. We have at most
$$\prod_{k \in K} |LT_k|^{|\widehat{l}^k_\alpha(\widehat{t})|} \leq \left(2^{4 \cdot |P_k|}\right)^{|\widehat{l}^k_\alpha(\widehat{t})|} = 2^{\left(\sum_{k \in K} |\widehat{l}^k_\alpha(\widehat{t})| \cdot 4 \cdot |P_k|\right)} \in 2^{O(m)}$$
different choices for the synchronisation partner and therefore at most $2^{O(m)}$ different synchronisation functions $\vartheta$.
4. We check whether $\widehat{t}$ and $\vartheta(N)$ have matching multisets of labels, i.e. $\widehat{l}(\widehat{t})(N) = l^\sharp_N(\vartheta(N))$ for all $N \in \mathcal{U}$.
5. We guess a mode $(\lambda, \rho)$. Since we consider safe Hornets the choice for $\lambda$ is unique, since there is at most one net-token on each place, which implies that whenever an event $\widehat{t}^\alpha[\vartheta]$ is enabled, then $\lambda$ is uniquely determined.
For the multiset $\rho$, we use the fact that for each place in the object net $N$ we have at most one token to distribute over all generated net-tokes. For each net $N$ we select one of the net-tokens generated in the postset, i.e. on the places $t^\bullet \cap d^{-1}(N)$.
Since $|t^\bullet \cap d^{-1}(N)| =: c$ is constant, we have at most $c^{|P_N|} \in 2^{O(|P_N|)} = 2^{O(m)}$ different modes.

All these choices need at most $2^{O(m)}$ bits of space.

We then check whether the event is enable, i.e. whether $\mu \xrightarrow{\widehat{t}^{\alpha}(\lambda,\rho)}$ holds.

1. We test $\alpha \sqsubseteq \mu$. This holds iff $\exists \mu'' : \mu = \alpha + \mu''$. Since $\alpha$ and $\mu$ are known, this can be tested by 'tagging' all addends from $\alpha$ in $\mu$, which needs at most $2^{O(m)}$ bits.
2. We evaluate the guard predicate $\widehat{G}(\widehat{t})$.
3. The resulting successor marking is defined as $\mu' = \mu - \lambda + \rho$. Here, $\mu - \lambda$ are those addends, that have not been tagged. Adding $\rho$ needs at most $2^{O(m)}$ bits.

Since we have at most $2^{2^{O(m)}}$ different reachable markings, the markings $\mu_0$, $\mu_1$, $\mu$, and $\mu'$ are storable in $2^{O(m)}$ bits. The counter $c$ is storable in $2^{O(m)}$ bits, too. Therefore, the whole algorithm needs at most $2^{O(m)}$ bits of space to decide reachability for eHornets. qed.

We use a well known fact about non-deterministic space complexity:

**Theorem 1 (Savitch).** *For each place constructible function $s(n) \geq \log(n)$ we have $NSpace(s(n)) \subseteq DSpace(s^2(n))$.*

Now we use the technique of Savitch to construct a deterministic algorithm from the non-determistic algorithm above.

**Theorem 2.** *The reachability problem $\mathsf{Reach}_{\mathsf{seH}}$ for safe* eHornets *can be solved within exponential space: $\mathsf{Reach}_{\mathsf{seH}} \in DSpace(2^{O(m)})$.*

*Proof.* We known by Lemma 1 that $\mathsf{Reach}_{\mathsf{seH}} \in NSpace(2^{O(m)})$.

From Savitch's Theorem we obtain that $\mathsf{Reach}_{\mathsf{seH}} \in DSpace\left(\left(2^{O(m)}\right)^2\right) = DSpace\left(2^{2 \cdot O(m)}\right) = DSpace\left(2^{O(m)}\right)$. qed.

## 4 Conclusion

In this paper we studied the subclass of safe eHornets. While reachability is PSpace-complete for safe Eos, we obtain that the reachability problem for safe eHornets requires at leats exponential space.

Our result here tightens a previous result for the upper bound of reachability. [12] establishes a *double* exponential space complexity using a simple reduction technique. Here, we have shown that exponential space is sufficient, i.e. upper and lower bound are equal.

We like to emphasise that the power of safe eHornets is only due to the transformations of the net-algebra, since all other sources of complexity (nesting, multiple tokens on a place, etc.) have been ruled out by the restriction of safeness and elementariness.

## References

1. Köhler, M., Rölke, H.: Properties of Object Petri Nets. In Cortadella, J., Reisig, W., eds.: International Conference on Application and Theory of Petri Nets 2004. Volume 3099 of Lecture Notes in Computer Science., Springer-Verlag (2004) 278–297
2. Köhler-Bußmeier, M., Heitmann, F.: On the expressiveness of communication channels for object nets. Fundamenta Informaticae **93** (2009) 205–219
3. Valk, R.: Object Petri nets: Using the nets-within-nets paradigm. In Desel, J., Reisig, W., Rozenberg, G., eds.: Advanced Course on Petri Nets 2003. Volume 3098 of Lecture Notes in Computer Science., Springer-Verlag (2003) 819–848
4. Reisig, W.: Petri nets and algebraic specifications. Theoretical Computer Science **80** (1991) 1–34
5. Hoffmann, K., Ehrig, H., Mossakowski, T.: High-level nets with nets and rules as tokens. In: Application and Theory of Petri Nets and Other Models of Concurrency. Volume 3536 of Lecture Notes in Computer Science., Springer-Verlag (2005) 268 – 288
6. Lomazova, I.: Nested petri nets for adaptive process modeling. In Avron, A., Dershowitz, N., Rabinovich, A., eds.: Pillars of Computer Science. Volume 4800 of Lecture Notes in Computer Science. Springer-Verlag (2008) 460–474
7. Köhler-Bußmeier, M.: Hornets: Nets within nets combined with net algebra. In Wolf, K., Franceschinis, G., eds.: International Conference on Application and Theory of Petri Nets (ICATPN'2009). Volume 5606 of Lecture Notes in Computer Science., Springer-Verlag (2009) 243–262
8. Köhler-Bußmeier, M., Heitmann, F.: Safeness for object nets. Fundamenta Informaticae **101** (2010) 29–43
9. Köhler-Bußmeier, M., Heitmann, F.: Liveness of safe object nets. Fundamenta Informaticae **112** (2011) 73–87
10. Köhler-Bußmeier, M.: On the complexity of safe, elementary Hornets. In: Proceedings of the International Workshop on Concurrency, Specification, and Programming (CS&P 2012). Volume 928., CEUR Workshop Proceedings (2012)
11. Lipton, R.J.: The reachability problem requires exponential space. Research Report 62, Dept. of Computer science (1976)
12. Köhler-Bußmeier, M., Heitmann, F.: Complexity results for elementary Hornets. In Colom, J.M., Desel, J., eds.: PETRI NETS 2013. Volume 7927 of Lecture Notes in Computer Science., Springer-Verlag (2013) 150–169
13. Ehrig, H., Mahr, B.: Fundamentals of algebraic Specification. EATCS Monographs on TCS. Springer-Verlag (1985)