A Survey of Elementary Object Systems
Part I: Decidability Results

Michael Köhler-Bußmeier

University of Hamburg, Department for Informatics
Vogt-Kölln-Straße 30, D-22527 Hamburg
koehler@informatik.uni-hamburg.de

Abstract. This contribution presents the formalism of Elementary Object Systems (Eos). Object nets are Petri nets which have Petri nets as tokens – an approach known as the nets-within-nets paradigm.
Since object nets in general are immediately Turing complete, we introduce the restricted class of elementary object nets which restrict the nesting of nets to the depth of two.
In this work we study the relationship of Eos to existing Petri net formalisms. It turns out that Eos are more powerful than classical p/t nets which is demonstrated by the fact that e.g. reachability and liveness become undecidable problems for Eos. Despite these undecidability results other properties can be extended to Eos using a monotonicity argument similar to that for p/t nets.
Also linear algebraic techniques, especially the theory of linear invariants and semiflows, can be extended in an appropriate way. The invariant calculus for Eos even enjoys the property of compositionality, i.e. invariants of the whole system can be composed of invariants of the object nets, which reduces the computational effort.
To obtain a finer level of insight we also study several subclasses. Among these variants the subclass of *generalised state machines* is worth mentioning since it combines the decidability of many theoretically interesting properties with a quite rich practical modelling expressiveness.
We also study safe Eos, a generalisation of safe p/t nets which are bounded systems with bound $b = 1$. Four different variants of safeness are studied. It turns out that variants are equivalent for p/t like Eos. While reachability and liveness remain undecidable for the two weaker classes of safe Eos, the two most strongest variants are restrictive enough to guarantee decidability. In fact, both problems are Pspace-complete.

Keywords: Petri nets, nets-within-nets, nets as tokens, object nets, safeness, reachability, liveness, boundedness

Abstract. Dieser Artikel präsentiert den Formalismus der *Elementaren Objektsysteme* (Eos). Objektnetze sind Petrinetze, die wiederum Petrinetze als Marken besitzen – ein Ansatz der auch als das Paradigma der *Netze-in-Netzen* bekannt ist.
Da Objektnetze im allgemeinen die Audrucksstäke von Turing-Maschinen besitzen, beschränken wir die Betrachtung hier auf die Klasse der elementaren Objektnetze, deren Schachtelungstiefe auf zwei Ebenen festgelegt ist.
Wir betrachten die Beziehungen von Eos zu anderen Petrinetzformalismen. Es zeigt sich, dass Eos ausdrucksmächtiger als die klassischen P/T-Netze sind, was sich darin äußert, dass insbesondere das Erreichbarkeits- und das Lebendigkeitsproblem für Eos unentscheidbar sind. Trotz dieser Unentscheidbarkeitsresultate können andere Eigenschaften auch für Eos als entscheidbar nachgewiesen werden, indem man die Monotonie der Schaltregel – wie sie bekanntermaßen für P/T-Netze gilt – auch für Eos nachweist.
Wir betrachten außerdem Techniken aus der linearen Algebra, hier insbesondere lineare Invarianten, die sich auch für Eos geignet definieren lassen. Wir zeigen, dass der Invariantenkalkül der Eos kompositional ist, d.h. man kann Invarianten des Gesamtsystems aus den Teil-Invarianten der Objektnetze erzeugen, was sinnvoll ist, um den Berechnungsaufwand zu reduzieren.
Um ein besseres Verständnis der Ausdrucksstärke der Objektnetze zu bekommen, betrachten wir auch strukturelle Teilklassen des Formalismus. Eine besonders interessante Teilklasse ist die der *generalised state machines*, da eine Vielzahl von Eigenschaften für diese Klasse entscheidbar ist und viele praktisch vorkommende Szenarien sich mit Netzen dieser Klasse modellieren lassen.
Wir betrachten auch noch sichere Eos – eine Verallgemeinerung sicherer P/T-Netze. P/T-Netze heißen sicher, wenn alle Plätze die Kapazitätsschranke $b = 1$ besitzen. Wir studieren hier vier verschiedene Varianten der Sicherheit in Eos. Wir zeigen, dass diese Varianten für P/T-artige Eos zueinander äquivalent sind, während sie für Eos eine Hierarchie bilden.
Während Erreichbarkeit und Lebendigkeit für die beiden schwächeren Formern der Sicherheit von Eos unentscheidbar bleiben, garantieren die beiden stärkeren Formen ihre Entscheidbarkeit. Man kann sogar konkreter zeigen, dass die Probleme in beiden Varianten Pspace-vollständige Probleme sind.

Schlagworte: Petrinetze, Netze-in-Netzen, Netze als Marken, Objektnetze, Sicherheit, Erreichbarkeit, Lebendigkeit, Beschränktheit

Table of Contents

1 Introduction

Object Systems are Petri nets which have Petri nets as tokens – an approach which is called the *nets-within-nets* paradigm, proposed by Valk [1991, 2003] for a two levelled structure and generalised in [Köhler and Rölke, 2003, 2004, Köhler-Bußmeier and Heitmann, 2009] for arbitrary nesting structures. The Petri nets that are used as tokens are called net-tokens. Net-tokens are tokens with internal structure and inner activity. This is different from place refinement, since tokens are transported while a place refinement is static. Net-tokens are some kind of *dynamic* refinement of states. The algebraic extension of objects nets – discussed in [Köhler-Bußmeier, 2009] – even allows operations on the net-tokens, like sequential or parallel composition. This is a concise way to express the self-modification of net-tokens at run-time in an algebraic setting.

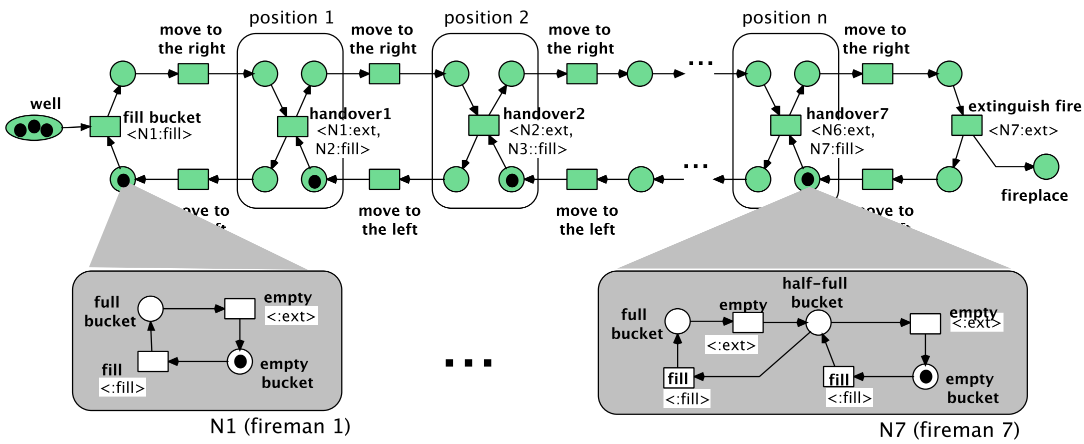


Fig. 1. The Bucket Chain as a Nested System

It is quite natural to use object nets to model mobility and mobile agents (cf. Köhler et al., 2003). Each place of the system net describes a location that hosts agents, which are net-tokens. Mobility can be modelled by moving the net-token from one place to another. This hierarchy forms a useful abstraction of the system: on a high level the agent system and on a lower level of the hierarchy the agent itself.

Without the viewpoint of nets as tokens, the modeller would have to encode the agent differently, e.g. as a data-type. This has the disadvantage, that the inner actions cannot be modelled directly, so, they have to be lifted to the system net, which seems quite unnatural.o By using nets-within-nets we can investigate the concurrency of the system and the agent in one model without loosing the abstraction needed.

Example 1. Figure 1 shows an object systems which models Carl Adam Petri's bucket chain scenario [Petri, 1979], where the fireman are mobile. In the bucket chain-example n firemen are standing in a row, each equipped with a bucket. A well is available for the leftmost fireman and the fire is at the rightmost place. So the leftmost fireman fills his bucket, while the rightmost extinguishes the fire. Neighboured firemen can handover buckets, so full buckets are handed over to right (to extinguish the fire) and empty ones to the left for refilling.[1] The topology (i.e. each fireman can only interact with his intermediate neighbour) introduces an interesting causal dependency structure[2]: The effect of exchanging buckets at a location being k steps away can be observed only when the whole system has moved k steps ahead.

Figure 1 shows the Petri net modelling the bucket chain with $n = 7$ firemen. Each fireman initially carries one empty bucket. The fireman man are mobile: They are net-tokens that move around in the system net which models the chain itself. The agents have different capabilities. For example the object net N_7 has a bucket with double capacity. At the well the object net N_1 fires the transition fill bucket which synchronises over the communication channel $\langle$:fill$\rangle$ with the transition fill of the object net. Then the object net N_1 moves to the position 1 firing the transition move to the right. At the position 1 N_1 synchronises with N_1 firing the transition handover which has two invocations of communication channels: The object net N_1 is synchronised over the channel $\langle$:ext$\rangle$ which changes its state to empty bucket. Synchronously, the object net N_2 is synchronised over the channel $\langle$:fill$\rangle$ which changes its state to full bucket. Thereafter the object net N_1 moves back to the well via the transition move to the left and the object net N_2 moves in the direction of the fire via the transition move to the right. Similarly for the other positions. So, the firemen transport full buckets to the right and the fire will be finally extinguished.

Among the wealth of research on defining mobile systems, in recent years a variety of formalisms have been introduced or adopted to cover mobility: The approaches can be roughly separated into process calculi and Petri net based approaches. The π-calculus [Milner et al., 1992], the Ambient calculus [Cardelli et al., 1999] and the Seal calculus [Vitek and Castagna, 1998] are just three of the more popular calculi. Approaches dealing with mobility and Petri nets are elementary object net systems [Valk, 1998, 2003], mobile nets [Busi, 1999], recursive nets [Haddad and Poitrenaud, 1999], minimal object nets [Kummer, 2000], nested nets [Lomazova, 2000], mobile predicate/transition nets [Xu and Deng, 2000], Reference nets [Kummer, 2002], PN^2 [Hiraishi, 2002], hypernets [Bednarczyk et al., 2004], object net systems [Köhler and Rölke, 2004, 2005, Köhler-Bußmeier and Heitmann, 2009], Mobile Systems [Lakos, 2005], AHO systems [Hoffmann et al., 2005], mobile object nets [Köhler and Farwer, 2006], adaptive workflow nets [Lo-

[1] This scenario has been introduced to study the causal dependencies of distributed cooperation. The scenario serves a similar purpose as the well known *Bankers-Problem* for deadlock-prevention in resource allocation systems (i.e. operating systems) or the *Dining Philosophers* for the study of fairness in distributed systems Peterson and Silberschatz, 1985.

[2] In the general research of Petri this causal dependency structure is closely related to Einstein's physical theory of relativity. This topic is studied in Petri's research of general net theory.

mazova et al., 2006], ν-Abstract Petri nets [Velardo and de Frutos-Escrig, 2008], and Hornets [Köhler-Bußmeier, 2009].

One central aim of this contribution is to compile existing results on special aspects of Eos together with some unpublished properties within one self-contained presentation. As a byproduct most proofs have been rewritten and shortened in a way appropriate for the newer results. The paper has the following structure: Section 2 recalls basic notations of Petri nets. Section 3 defines elementary object systems (Eos). Section 4 provides a short overview of related nets-within-nets formalisms. Section 5 investigates the invariance calculus for Eos and demonstrates its compositionality. Section 6 studies decidability problems for Eos, namely: reachability, liveness and boundedness. Section 7 studies the same problems for Conservative Eos which are restricted in a way that object nets are copied or fused but never created or destroyed. It will turn out that this restriction regains the monotonicity of the firing rule which is lost in the general case. Section 8 compares Eos with a reference semantics based on p/t nets and introduces a sub class, called Generalised State Machines, which is of practical interest, because models of this class corresponds to scenarios related to physical entities. Section 9 studies Eos the properties of Eos which respect certain bounds on their markings. It turns out that not all bounds guarantee decidability of the standard problems or finite state spaces, that are usually assumed by model-checking techniques. In generalisation of p/t nets we consider the property of safeness for Eos, i.e. a property which restricts the number of tokens on each place to at most one.

2 Preliminaries

The definition of Petri nets relies on the notion of multisets. A multiset $\mathbf{m}$ on the set D is a mapping $\mathbf{m} : D \to \mathbb{N}$. Multisets are generalisations of sets in the sense that every subset of D corresponds to a multiset $\mathbf{m}$ with $\mathbf{m}(d) \leq 1$ for all $d \in D$. The notation is used for sets as well as for multisets. The meaning will be apparent from its use. Multiset addition $\mathbf{m}_1, \mathbf{m}_2 : D \to \mathbb{N}$ 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 $(\mathbf{m}_1 - \mathbf{m}_2)(d) := \max(\mathbf{m}_1(d) - \mathbf{m}_2(d), 0)$. We use common notations for the cardinality of a multiset $|\mathbf{m}| := \sum_{d \in D} \mathbf{m}(d)$ and multiset ordering $\mathbf{m}_1 \leq \mathbf{m}_2$, where the partial order $\leq$ is defined by $\mathbf{m}_1 \leq \mathbf{m}_2 \iff \forall d \in D : \mathbf{m}_1(d) \leq \mathbf{m}_2(d)$. A multiset $\mathbf{m}$ is finite if $|\mathbf{m}| < \infty$. The set of all finite multisets over the set D is denoted $MS(D)$. The set $MS(D)$ naturally forms a monoid with multiset addition $+$ and the empty multiset $\mathbf{0}$. Multisets can be identified with the commutative monoid structure $(MS(D), +, 0)$. Multisets are the free commutative monoid over D since every multiset has the unique representation in the form $\mathbf{m} = \sum_{d \in D} \mathbf{m}(d) \cdot d$, where $\mathbf{m}(d)$ denotes the multiplicity of d. Multisets can be represented as a formal sum in the form $\mathbf{m} = \sum_{i=1}^{n} x_i$, where $x_i \in D$.

Any mapping $f : D \to D'$ can be extended to a homomorphism $f^\sharp : MS(D) \to MS(D')$ on multisets: $f^\sharp\left(\sum_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} f(x_i)$. This includes the special case $f^\sharp(\mathbf{0}) = \mathbf{0}$. We simply write f to denote the mapping $f^\sharp$. The notation is in accordance with the set-theoretic notation $f(A) = \{f(a) \mid a \in A\}$.

Definition 1. *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:* $\mathbf{m} \in MS(P)$*. A p/t net with initial marking* $\mathbf{m}$ *is denoted* $N = (P, T, \mathbf{pre}, \mathbf{post}, \mathbf{m})$*.*

We use the usual notations for nets like ${}^\bullet x$ for the set of predecessors and $x^\bullet$ for the set of successors for a node $x \in (P \cup T)$.

A transition $t \in T$ of a p/t net N is enabled in marking $\mathbf{m}$ iff $\forall p \in P : \mathbf{m}(p) \geq \mathbf{pre}(t)(p)$ holds. The successor marking when firing t is $\mathbf{m}'(p) = \mathbf{m}(p) - \mathbf{pre}(t)(p) + \mathbf{post}(t)(p)$ for all $p \in P$. Using multiset notation enabling is expressed by $\mathbf{m} \geq \mathbf{pre}(t)$ and the successor marking is $\mathbf{m}' = \mathbf{m} - \mathbf{pre}(t) + \mathbf{post}(t)$. We denote the enabling of t in marking $\mathbf{m}$ by $\mathbf{m} \xrightarrow[N]{t}$. Firing of t is denoted by $\mathbf{m} \xrightarrow[N]{t} \mathbf{m}'$. The net N is omitted if it is clear from the context.

Firing is extended to sequences $w \in T^*$ in the obvious way:

(i) $\mathbf{m} \xrightarrow{\epsilon} \mathbf{m}$;

(ii) If $\mathbf{m} \xrightarrow{w} \mathbf{m}'$ and $\mathbf{m}' \xrightarrow{t} \mathbf{m}''$ hold, then we have $\mathbf{m} \xrightarrow{wt} \mathbf{m}''$.

We write $\mathbf{m} \xrightarrow{*} \mathbf{m}'$ whenever there is some $w \in T^*$ such that $\mathbf{m} \xrightarrow{w} \mathbf{m}'$ holds.

The set of reachable markings is $RS(\mathbf{m}_0) : \{\mathbf{m} \mid \exists w \in T^* : \mathbf{m}_0 \xrightarrow{w} \mathbf{m}\}$.

3 Elementary Object Systems

An elementary object system (EOS) is composed of a system net, which is a p/t net $\widehat{N} = (\widehat{P}, \widehat{T}, \mathbf{pre}, \mathbf{post})$ and a set of object nets $\mathcal{N} = \{N_1, \ldots, N_n\}$, which are p/t nets given as $N = (P_N, T_N, \mathbf{pre}_N, \mathbf{post}_N)$. In extension we assume that all sets of nodes (places and transitions) are pairwise disjoint. Moreover we assume $\widehat{N} \notin \mathcal{N}$. We assume the existence of the object net $\bullet \in \mathcal{N}$ which has no places and no transitions and is used to model anonymous, so called black tokens.

The system net places are typed by the mapping $d : \widehat{P} \to \mathcal{N}$ with the meaning, that the place $\widehat{p}$ of the system net contains net-tokens of the object net type N if $d(\widehat{p}) = N$.[3] No place of the system net is mapped to the system net itself since $\widehat{N} \notin \mathcal{N}$.

Nested Markings Since the tokens of an EOS are instances of object nets a *marking* $\mu \in \mathcal{M}$ of an EOS *OS* is a *nested* multiset.

A marking of an EOS *OS* is denoted $\mu = \sum_{k=1}^{|\mu|} (\widehat{p}_k, M_k)$, where $\widehat{p}_k$ is a place in the system net and M_k is the marking of the net-token of type $d(\widehat{p}_k)$. To emphasise the nesting, markings are also denoted as $\mu = \sum_{k=1}^{|\mu|} \widehat{p}_k[M_k]$. Tokens of the form $\widehat{p}[\mathbf{0}]$ and $d(\widehat{p}) = \bullet$ are abbreviated as $\widehat{p}[]$.

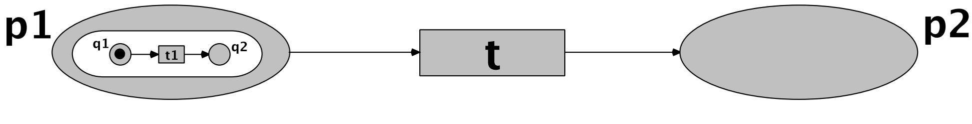
The set of all markings which are syntactically consistent with the typing d is denoted $\mathcal{M}$ (Here $d^{-1}(N) \subseteq \widehat{P}$ is the set of system net places of the type N):

$$\mathcal{M} := MS\left(\bigcup_{N \in \mathcal{N}} \left(d^{-1}(N) \times MS(P_N)\right)\right) \tag{1}$$

We define the partial order $\sqsubseteq$ on nested multisets by setting $\mu_1 \sqsubseteq \mu_2$ iff $\exists \mu : \mu_2 = \mu_1 + \mu$.

Events Analogously to markings, which are nested multisets μ, the events of an EOS are also nested. An EOS allows three different kinds of events (cf. the following EOS).

1. System-autonomous: The system net transition t fires autonomously which moves the net-token from p_1 to p_2.
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 p_1.
3. Synchronisation: The system net transition t fires synchronously with t_1 in the object-net. Whenever synchronisation is demanded then autonomous actions are forbidden.



[3] In the following the terms *(marked) object net* and *net-token* are used almost interchangeable. We use the term *net-token* whenever we like to emphasise the aspect that the marked object net is a token of the system net.

These three kinds of events can be described in a uniform way, namely as synchronisations: $\widehat{t}[\vartheta]$, where $\widehat{t}$ is the transition that fires in the system net and $\vartheta(N)$ is a multiset of its transitions (i.e. $\vartheta(N) \in MS(T_N)$ for each object net $N \in \mathcal{N}$), which have to fire synchronously with $\widehat{t}$.[4]

Obviously system-autonomous events are a special case of synchronous events, where $\vartheta(N) = \mathbf{0}$ for all object nets N. To describe object-autonomous events we assume the set of *idle transitions* $\{id_{\widehat{p}} \mid \widehat{p} \in \widehat{P}\}$ to be included in the set of system net transitions $\widehat{T}$, where $id_{\widehat{p}}$ formalises object-autonomous firing on the place $\widehat{p}$:

1. Each idle transitions $id_{\widehat{p}}$ has $\widehat{p}$ as its side condition: $\mathbf{pre}(id_{\widehat{p}}) = \mathbf{post}(id_{\widehat{p}}) := \widehat{p}$.
2. Each idle transition $id_{\widehat{p}}$ synchronises only with one transition from $N = d(\widehat{p})$:

$$\forall \widehat{\tau}[\vartheta] \in \Theta : \widehat{\tau} = id_{\widehat{p}} \Longrightarrow \forall N \in \mathcal{N} : |\vartheta(N)| \leq 1 \wedge \\ (\vartheta(N) \neq \mathbf{0} \iff N = d(\widehat{p}))$$

With these idle transitions all three kinds of events are described as a synchronisation event $\widehat{\tau}[\vartheta]$, where $\widehat{\tau}$ is either a "real" transition $\widehat{t}$ or $id_{\widehat{p}}$ for some $\widehat{p}$.

Definition 2 (EOS). *An elementary object system (*Eos*) is a tuple $OS = (\widehat{N}, \mathcal{N}, d, \Theta, \mu_0)$ such that:*

1. *$\widehat{N}$ is a p/t net, called the* system net.
2. *$\mathcal{N}$ is a finite set of disjoint p/t nets, called* object nets.
3. *$d : \widehat{P} \to \mathcal{N}$ is the typing of the system net places.*
4. *Θ is the set of events.*
5. *$\mu_0 \in \mathcal{M}$ is the initial marking.*

A typing is called conservative *iff for each place in the preset of a system net transition $\widehat{t}$ such that $d(\widehat{p}) \neq \bullet$ there is place in the postset being of the same type: $(d(^{\bullet}\widehat{t}) \cup \{\bullet\}) \subseteq (d(\widehat{t}^{\bullet}) \cup \{\bullet\})$. An* Eos *is* conservative *iff its typing d is.*

An Eos *is* p/t-like *iff it has only places for black tokens: $d(\widehat{P}) = \{\bullet\}$.*

We name special properties of Eos:

- An Eos is *minimal* iff it has exactly one "real" object net: $|\mathcal{N} \setminus \{\bullet\}| = 1$.
- An Eos is *pure* iff it has no places for black tokens: $d^{-1}(\bullet) = \emptyset$.
- An Eos is *unary* iff it is pure and minimal.

It is possible to define synchronisations which are never enabled. To prevent the most obvious cases we require that each event $\theta = \widehat{\tau}[\vartheta]$ has to respect the following:

---

[4] In the graphical representation the events are generated by transition inscriptions. For each object net $N \in \mathcal{N}$ a system net transition $\widehat{t}$ is labelled with a multiset of channels $\widehat{l}(\widehat{t})(N) = ch_1 + \cdots + ch_n$ which depicted as $\langle N{:}ch_1, N{:}ch_2, \ldots \rangle$. Similarily, an object net transition t may be labelled with a channel $l_N(t) = ch$ – depicted as $\langle {:}ch \rangle$ whenver there is such a label. We obtain an event $\widehat{t}[\vartheta]$ by setting $\vartheta(N) := t_1 + \cdots + t_n$ to be any transition multiset such that labels match: $l_N(t_1) + \cdots + l_N(t_n) = \widehat{l}(\widehat{t})(N)$.

1. Let $\mathcal{N}(t) := d(^\bullet\widehat{t} \cup \widehat{t}^\bullet)$ denote the set of objects nets in the location of $\widehat{t}$. We assume that the system net synchronises only with N if there is a place attached to $\widehat{t}$ that may contain tokens of the object net type N:

$$N \notin \mathcal{N}(t) \Longrightarrow \vartheta(N) = \mathbf{0}$$

   Without this property, the system net transition $\widehat{t}$ would always be disabled since there is no chance to synchronise since there is no such object net as a net-token.
2. Since it is not possible to synchronise with black tokens, we assume $\vartheta(N) = \mathbf{0}$ for $N = \bullet$.
   Without this property, the event would always be disabled since $N = \bullet$ has no transitions by definition.

Example 2. Figure 2 shows an Eos with the system net $\widehat{N}$ and the object nets $\mathcal{N} = \{N_1, N_2\}$. The system has four net-tokens: two on place p_1 and one on p_2 and p_3 each. The net-tokens on p_1 and p_2 share the same net structure, but have independent markings.

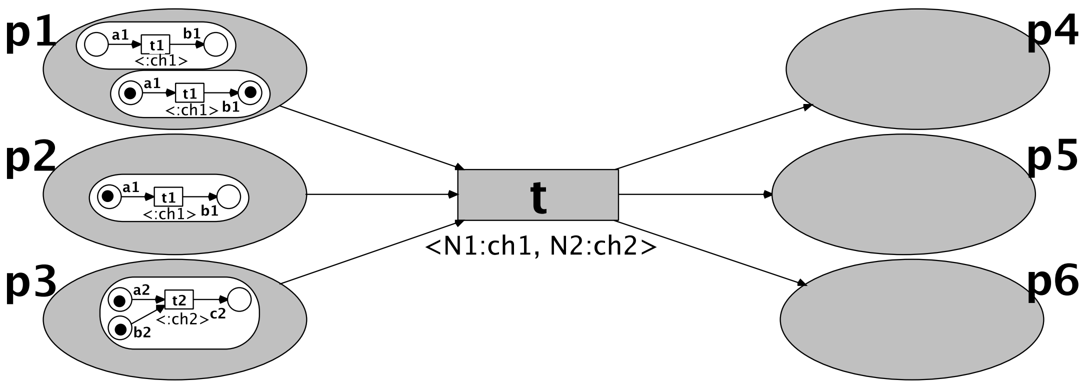


Fig. 2. An Elementary Object Net System

The system net $\widehat{N} = (\widehat{P}, \widehat{T}, \mathbf{pre}, \mathbf{post})$, where $\widehat{P} = \{p_1, \ldots, p_6\}$ and $\widehat{T} = \{t\}$.
The object net $N_1 = (P_1, T_1, \mathbf{pre}_1, \mathbf{post}_1)$ with $P_1 = \{a_1, b_1\}$ and $T_1 = \{t_1\}$.
The object net $N_2 = (P_2, T_2, \mathbf{pre}_2, \mathbf{post}_2)$ with $P_2 = \{a_2, b_2, c_2\}$ and $T_2 = \{t_2\}$.
The typing is $d(p_1) = d(p_2) = d(p_4) = N_1$ and $d(p_3) = d(p_5) = d(p_6) = N_2$.
We have the channels ch_1 and ch_2. The labelling function of the system net $\widehat{l}$ is defined by $\widehat{l}(t)(N_1) = ch_1$ and $\widehat{l}(t)(N_2) = ch_2$.
The labelling l_{N_1} of the first object net is defined by setting $l_{N_1}(t_1) = ch_1$. Similarly, l_{N_2} is defined by $l_{N_2}(t_2) = ch_2$.
There is only one event: $\Theta = \{t[N_1 \mapsto t_1, N_2 \mapsto t_2]\}$.
The initial marking has two net-tokens on p_1, one on p_2, and one on p_3:

$$\mu = p_1[a_1 + b_1] + p_1[\mathbf{0}] + p_2[a_1] + p_3[a_2 + b_2]$$

Note that for Figure 2 the structure is the same for the three net-tokens on p_1 and p_2 but the net-tokens' markings are different.

Communication Channels In the following we define the transition labels in detail. The set of all possible synchronisations is constructed via channel inscriptions. We assume a fixed set of channels $C = \bigcup_{N \in \mathcal{N}} C_N$ for disjoint C_N. The transitions in an Eos are labelled with synchronisation channels with the intention that a system net's transition synchronises with the object nets' transitions that have corresponding channels.

- For each transition $\widehat{t}$ the function $\widehat{l}$ assigns to each object nets N a multiset of channels: $\widehat{l}(\widehat{t})(N) \in MS(C_N)$.[5] Whenever $\widehat{l}(\widehat{t})(N) = \mathbf{0}$ for the object net N, then $\widehat{t}$ does not synchronise with N.
  Since it is not possible to synchronise with black tokens, we assume $\widehat{l}(\widehat{t})(N) = \mathbf{0}$ for $N = \bullet$.
- The partial function l_N assigns to some transitions t of the object net N a channel, i.e. $l_N(t) \in C_N$ whenever defined. If $l_N(t)$ is undefined, then t fires without synchronisation, i.e. autonomously.
  For technical simplification we turn the partial function into a total one using the fresh "channels" $\bot_N$: Whenever $l_N(t)$ is undefined, we set $l_N(t) := \bot_N$.

In the graphical representation the synchronisation labelling is expressed by transition inscriptions in the form $\langle N_1 : \widehat{l}(\widehat{t})(N_1), \ldots, N_k : \widehat{l}(\widehat{t})(N_k)\rangle$ for the system net (whenever the label is not the empty multiset) and in the form $\langle : l_N(t)\rangle$ for the object nets for (whenever defined).

Define $\mathcal{N}(t) := d({}^\bullet\widehat{t} \cup \widehat{t}^\bullet)$. We assume that the system net's labelling $\widehat{l}(\widehat{t})(N)$ does not have a channel inscription from the set C whenever there is no place attached to $\widehat{t}$ which can contain tokens of the object net N:

$$N \notin \mathcal{N}(t) \Longrightarrow \widehat{l}(\widehat{t})(N) = \mathbf{0} \tag{2}$$

Otherwise, the system net transition $\widehat{t}$ would always be disabled since there is no chance to synchronise since there is no such object net as a net-token.

The labelling introduces three kinds of events:

1. System-autonomous firing: The transition $\widehat{t}$ of the system net fires autonomously, whenever $\widehat{l}(\widehat{t})(N) = \mathbf{0}$ for all $N \in \mathcal{N}$.
2. Synchronised firing: There is at least one object net that has to be synchronised, i.e. there is a N such that $|\widehat{l}(\widehat{t})(N)| > 0$.
3. Object-autonomous firing: An object net transition t fires autonomously whenever $l(t)$ is undefined.

[5] In previous formalisation of the synchronisation we bounded the number of channels for each object net by one. The main motivation for the restriction was to obtain a shorter presentation. In this presentation we dropped the restriction, since it has no influence on the results proven here.

These three kinds of events can be reduced to the case of a synchronisation, where a system net transition has a synchronisation partner in each object net. This normal form is obtained by synchronsing with some *idle transitions* which have no visible effect: We add the set of idle transitions $\{id_{\widehat{p}} \mid \widehat{p} \in \widehat{P}\}$ to the set of system net transitions to express object-atonomous events and define $\mathbf{pre}(id_{\widehat{p}}) = \mathbf{post}(id_{\widehat{p}}) = \widehat{p}$, i.e. $\widehat{p}$ is a side condition of $id_{\widehat{p}}$.

We extend the labelling to idle transitions:

$$\widehat{l}(id_{\widehat{p}})(N) = \begin{cases} \bot_N & \text{if } d(\widehat{p}) = N \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

The meaning is that an object net $d(\widehat{p}) = N$ that fires autonomously within the place $\widehat{p}$ has the idle transition $id_{\widehat{p}}$ as its partner in the system net.

The synchronisation labelling generates the set of system events Θ: An event is a pair – denoted $\theta = \widehat{\tau}[\vartheta]$ in the following, where $\widehat{\tau}$ is either a real transition $\widehat{t}$ or $id_{\widehat{p}}$ for some $\widehat{p}$. The function ϑ maps each object net to a multiset of its transitions. A pair $\widehat{\tau}[\vartheta]$ is an event whenever the system net transition $\widehat{\tau}$ fires synchronously with all the object net transitions $\vartheta(N), N \in \mathcal{N}(t)$, which is the case when the labels match, i.e. $\widehat{l}(\widehat{t})(N) = l^\sharp_N(\vartheta(N))$ for all $N \in \mathcal{N}(t)$. The set of all events is:

$$\Theta_l := \left\{\widehat{\tau}[\vartheta] \mid \forall N \in \mathcal{N}(t) : \widehat{l}(\widehat{t})(N) = l^\sharp_N(\vartheta(N))\right\} \tag{3}$$

A special case for the mapping ϑ is the idle map ϑ_{id} which is defined $\vartheta_{id}(N) = \mathbf{0}$ for all $N \in \mathcal{N}$. The idle map models system-autonomous events which have the form $\widehat{t}[\vartheta_{id}]$.

For object-autonomous events $id_{\widehat{p}}[\vartheta]$ the labelling guarantees $\vartheta(N') = \mathbf{0}$ except for the object net $N = d(\widehat{p})$.

3.1 Firing Rule

Let μ be a marking of an Eos. The projection Π^1 on the first component abstracts away the substructure of all net-tokens:

$$\Pi^1\left(\sum\nolimits_{k=1}^{|\mu|} \widehat{p}_k[M_k]\right) := \sum\nolimits_{k=1}^{|\mu|} \widehat{p}_k \tag{4}$$

The projection Π^2_N on the second component is the abstract marking of all net-tokens of the type $N \in \mathcal{N}$ ignoring their local distribution within the system net.

$$\Pi^2_N\left(\sum\nolimits_{k=1}^{|\mu|} \widehat{p}_k[M_k]\right) := \sum\nolimits_{k=1}^{|\mu|} \mathbf{1}_N(\widehat{p}_k) \cdot M_k \tag{5}$$

where the indicator function $\mathbf{1}_N : \widehat{P} \to \{0,1\}$ is $\mathbf{1}_N(\widehat{p}) = 1$ iff $d(\widehat{p}) = N$. Note that $\Pi^2_N(\mu)$ results in a marking of the object net N.

A system event $\widehat{\tau}[\vartheta]$ removes net-tokens together with their individual internal markings. Firing the event replaces a nested multiset $\lambda \in \mathcal{M}$ that is part of the current marking μ, i.e. $\lambda \sqsubseteq \mu$, by the nested multiset ρ. Therefore the successor marking is $\mu' := (\mu - \lambda) + \rho$. The enabling condition is expressed by the *enabling predicate* ϕ_{OS} (or just ϕ whenever OS is clear from the context):

$$\begin{aligned}\phi(\widehat{\tau}[\vartheta], \lambda, \rho) \iff\; & \Pi^1(\lambda) = \mathbf{pre}(\widehat{\tau}) \wedge \Pi^1(\rho) = \mathbf{post}(\widehat{\tau}) \wedge \\ & \forall N \in \mathcal{N} : \Pi^2_N(\lambda) \geq \mathbf{pre}_N(\vartheta(N)) \wedge \\ & \forall N \in \mathcal{N} : \Pi^2_N(\rho) = \Pi^2_N(\lambda) - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N))\end{aligned} \tag{6}$$

With $\widehat{M} = \Pi^1(\lambda)$ and $\widehat{M}' = \Pi^1(\rho)$ as well as $M_N = \Pi^2_N(\lambda)$ and $M'_N = \Pi^2_N(\rho)$ for all $N \in \mathcal{N}$ the predicate ϕ has the following meaning:

1. The first conjunct expresses that the system net multiset $\widehat{M}$ corresponds to the pre-condition of the system net transition $\widehat{t}$, i.e. $\widehat{M} = \mathbf{pre}(\widehat{t})$.
2. In turn, a multiset $\widehat{M}'$ is produced, that corresponds with the post-set of $\widehat{t}$.
3. An object net transition τ_N is enabled if the combination M_N of the markings net-tokens of type N enable it, i.e. $M_N \geq \mathbf{pre}_N(\vartheta(N))$.
4. The firing of $\widehat{\tau}[\vartheta]$ must also obey the *object marking distribution condition* which is essential for the formulation of linear invariants: $M'_N = M_N - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N))$, where $\mathbf{post}_N(\vartheta(N)) - \mathbf{pre}_N(\vartheta(N))$ is the effect of the object net's transition on the net-tokens.

Note that (1) and (2) assures that only net-tokens relevant for the firing are included in λ and ρ. Conditions (3) and (4) allows for additional tokens in the net-tokens.

For system-autonomous events $\widehat{t}[\vartheta_{id}]$ the enabling predicate ϕ can be simplified further. We have $\mathbf{pre}_N(id_N) = \mathbf{post}_N(id_N) = \mathbf{0}$. This ensures $\Pi^2_N(\lambda) = \Pi^2_N(\rho)$, i.e. the sum of markings in the copies of a net-token is preserved w.r.t. each type N. This condition ensures the existence of linear invariance properties (cf. Theorem 3).

Analogously, for an object-autonomous event we have an idle-transition $\widehat{\tau} = id_{\widehat{p}}$ for the system net and the first and the second conjunct is: $\Pi^1(\lambda) = \mathbf{pre}(\widehat{t}) = \widehat{p} = \mathbf{post}(\widehat{t}) = \Pi^1(\rho)$. So, there is an addend $\lambda = \widehat{p}[M]$ in μ with $d(\widehat{p}) = N$ and M enables $t_N := \vartheta(N)$.

Definition 3 (Firing Rule). *Let OS be an* EOS *and $\mu, \mu' \in \mathcal{M}$ markings. The event $\widehat{\tau}[\vartheta]$ is enabled in μ for the mode $(\lambda, \rho) \in \mathcal{M}^2$ iff $\lambda \sqsubseteq \mu \wedge \phi(\widehat{\tau}[\vartheta], \lambda, \rho)$ holds.*

An event $\widehat{\tau}[\vartheta]$ that is enabled in μ for the mode (λ, ρ) can fire: $\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu'$. The resulting successor marking is defined as $\mu' = \mu - \lambda + \rho$.

We write $\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta]} \mu'$ whenever $\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu'$ for some mode (λ, ρ).

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 (λ, ρ) to formulate the firing of $\widehat{\tau}[\vartheta]$ in a functional way.

Example 3. Consider the EOS of Figure 2 again. The current marking μ of the EOS enables $t[N_1 \mapsto t_1, N_2 \mapsto t_2]$ in the mode (λ, ρ), where

$$\begin{aligned}
\mu &= p_1[\mathbf{0}] + p_1[a_1 + b_1] + p_2[a_1] + p_3[a_2 + b_2] = p_1[\mathbf{0}] + \lambda \\
\lambda &= p_1[a_1 + b_1] + p_2[a_1] + p_3[a_2 + b_2] \\
\rho &= p_4[a_1 + b_1 + b_1] + p_5[\mathbf{0}] + p_6[c_2]
\end{aligned}$$

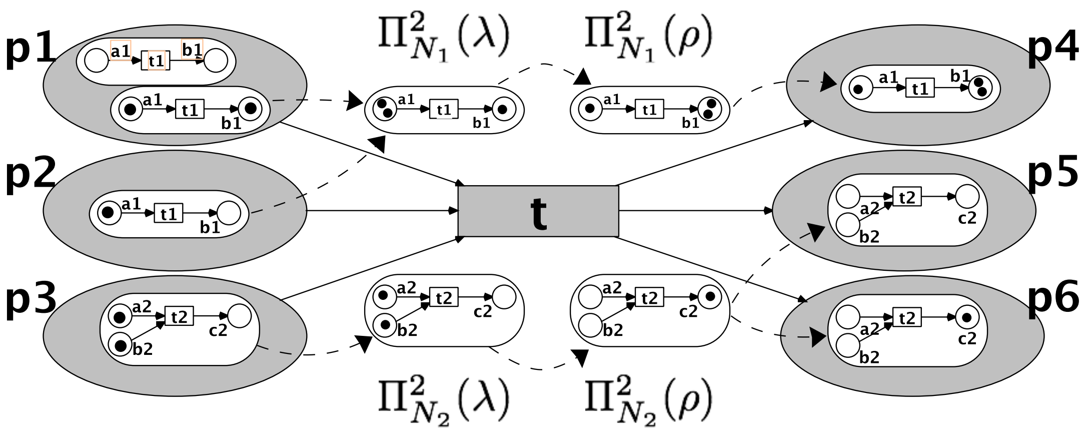


Fig. 3. The EOS of Figure 2 illustrating the projections $\Pi^2_N(\lambda)$ and $\Pi^2_N(\rho)$

The net-tokens' markings are added by the projections Π^2_N resulting in the markings $\Pi^2_N(\lambda)$. The sub-synchronisation generate $\Pi^2_N(\rho)$. (The results are shown above and below the transition t.) After the synchronisation we obtain the successor marking μ' with net-tokens on p_4, p_5, and p_6 as shown in the Figure 3:

$$\begin{aligned}
\mu' &= (\mu - \lambda) + \rho = p_1[\mathbf{0}] + \rho \\
&= p_1[\mathbf{0}] + p_4[a_1 + b_1 + b_1] + p_5[\mathbf{0}] + p_6[c_2]
\end{aligned}$$

3.2 Projection Equivalence

In the following we relate those nested multisets, that coincide in their projections. The projection of a marking μ is defined as follows:

$$\Pi(\mu) := (\Pi^1(\mu), (\Pi^2_N(\mu))_{N \in \mathcal{N}}) \tag{7}$$

Obviously, there are several markings μ with the same projection, i.e. μ is not uniquely defined by $\Pi(\mu)$. The nested multisets, that coincide in their projections give rise to the equivalence $\cong \; \subseteq \mathcal{M}^2$, called *projection equivalence* defined by:

$$\begin{aligned}
\alpha \cong \beta :\iff\; & \Pi(\alpha) = \Pi(\beta) \\
\iff\; & \Pi^1(\alpha) = \Pi^1(\beta) \wedge \forall N \in \mathcal{N} : \Pi^2_N(\alpha) = \Pi^2_N(\beta)
\end{aligned} \tag{8}$$

The relation $\alpha \cong \beta$ abstracts from the location, i.e. the concrete net-token, in which a object net's place p is marked as long as it is present in α and β. For example, for $d(\widehat{p}) = d(\widehat{p}')$ we have

$$\widehat{p}[p_1 + p_2] + \widehat{p}'[p_3] \cong \widehat{p}[p_3 + p_2] + \widehat{p}'[p_1]$$

which means that $\cong$ allows the tokens p_1 and p_3 to change their locations (i.e. between $\widehat{p}$ and $\widehat{p}'$).

Since an event collects all relevant object nets of the firing mode and combines them to one "virtual object net" that is only present at the moment of firing, the location of the object nets' tokens is irrelevant and can be ignored. These virtual object nets $\Pi^2_N(\lambda)$ are also show in the example of Figure 2. This invariance can be expressed as follows:

Lemma 1. *The enabling predicate is invariant with respect to the relation* $\cong$*:*

$$\phi(\widehat{\tau}[\vartheta], \lambda, \rho) \iff (\forall \lambda', \rho' : \lambda' \cong \lambda \wedge \rho' \cong \rho \Longrightarrow \phi(\widehat{\tau}[\vartheta], \lambda', \rho'))$$

Proof. From the definition of ϕ one can see that the firing mode (λ, ρ) is used only via its projection by Π. Since $\lambda' \cong \lambda, \rho' \cong \rho$ expresses equality modulo projection the predicate ϕ cannot distinguish between λ' and λ, resp. ρ' and ρ. □

As an immediate consequence of Lemma 1 we obtain the invariance of the firing rule with respect to projection equivalent modes.

Theorem 1. *Let OS be an* Eos *and μ a marking. The event $\widehat{\tau}[\vartheta]$ is enabled in the mode (λ, ρ) iff it is enabled in the mode (λ', ρ') such that $\lambda' \cong \lambda \wedge \rho' \cong \rho$.*

3.3 Properties of the Firing Rule

Of course there are a lot different possible candidates for the firing rule. In fact there infinitely many, since there is a lot of freedom how to distribute the net-tokens' markings when there are several outgoing arcs in the system net. Valk [2003] discusses three basic variants, called reference semantics, value semantics, and copy semantics. Reference semantics interprets net-tokens as pointers to object nets. This semantics can be equally expressed as a p/t net (cf. definition 7). Value semantics is the semantics presented in this paper. Copy semantics is a variant of value semantics, where the net-tokens' markings are not distributed over the outgoing net-tokens but copied. From the modelling point of view each of the semantics has its own pro and cons. From a more theoretical point of value semantics is a special one since it allows to reinterpret every firing sequence also with respect to reference semantics – as formulated in Theorem 10. One can even show that value semantics is the only one with this property (cf. Köhler, 2004 for details).

In the following we discuss two aspects that indicate that the firing rule has nice properties, namely reversibility and the fact that Eos are a canonical extension of p/t nets.

Reversibility A basic property of Petri nets is that their firing rule is symmetric in time, i.e. whenever all arcs are reversed then we can fire backwards: This is expressed by the reversed net $N^{rev} = (P, T, \mathbf{pre}^{rev}, \mathbf{post}^{rev})$ where $\mathbf{pre}^{rev} := \mathbf{post}$ and $\mathbf{post}^{rev} := \mathbf{pre}$ which is obtained from $N = (P, T, \mathbf{pre}, \mathbf{post})$ by dualising the effect. Symmetry in time is expressed as:

$$m_1 \xrightarrow[N]{t} m_2 \iff m_2 \xrightarrow[N^{rev}]{t} m_1$$

This property holds also for Eos. Given $OS = (\widehat{N}, \mathcal{N}, d, l)$ we define the *reverse* Eos as $OS^{rev} = (\widehat{N}^{rev}, \mathcal{N}^{rev}, d, l)$, where $\mathcal{N}^{rev} = \{N^{rev} \mid N \in \mathcal{N}\}$.

Lemma 2. *Let OS be an* Eos. *The enabling predicate is reversible:*

$$\phi_{OS}(\widehat{\tau}[\vartheta], \lambda, \rho) \iff \phi_{OS^{rev}}(\widehat{\tau}[\vartheta], \rho, \lambda)$$

Proof. Since $(OS^{rev})^{rev} = OS$ holds, it is sufficient to show only one implication.

Assume that $\phi(\widehat{\tau}[\vartheta], \lambda, \rho)$ holds with $\Pi(\lambda) = (\widehat{M}, (M_N)_{N\in\mathcal{N}})$ and $\Pi(\rho) = (\widehat{M'}, (M'_N)_{N\in\mathcal{N}})$.

We have $\widehat{M'} = \mathbf{post}(\widehat{\tau}) = \mathbf{pre}^{rev}(\widehat{\tau})$ and $\widehat{M} = \mathbf{pre}(\widehat{\tau}) = \mathbf{post}^{rev}(\widehat{\tau})$ which shows the first and the second conjunct of ϕ.

With $M_N \geq \mathbf{pre}_N(\vartheta(N))$ we can deduce the third conjunct of $\phi_{OS^{rev}}$:

$$\begin{aligned} & M'_N = M_N - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N)) \\ \implies & M'_N \geq \mathbf{post}_N(\vartheta(N)) = \mathbf{pre}^{rev}_N(\vartheta(N)) \end{aligned}$$

The fourth conjunct holds since:

$$\begin{aligned} & M'_N = M_N - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N)) \\ \iff & M_N = M'_N - \mathbf{post}_N(\vartheta(N)) + \mathbf{pre}_N(\vartheta(N)) \\ \iff & M_N = M'_N - \mathbf{pre}^{rev}_N(\vartheta(N)) + \mathbf{post}^{rev}_N(\vartheta(N)) \end{aligned}$$

Therefore, $\phi_{OS^{rev}}(\widehat{\tau}[\vartheta], \rho, \lambda)$ holds. □

This implies reversibility for Eos.

Proposition 1. *Let OS be an* Eos. *Firing is reversible:*

$$\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu' \iff \mu' \xrightarrow[OS^{rev}]{\widehat{\tau}[\vartheta](\rho,\lambda)} \mu$$

Proof. Since the enabling predicate ϕ is reversible (Lem. 2), it remains to show that the resulting successor marking μ' activates the event, i.e. that $\rho \sqsubseteq \mu'$. But this follows from the condition $\lambda \sqsubseteq \mu$ in the firing rule which implies: $\mu' = \mu - \lambda + \rho \geq \rho$. □

EOS as a Canonical Extension of P/T Nets EOS are a canonical extension of p/t nets in two ways: The behaviour of the system net in the EOS when ignoring the net-tokens structure cannot be distinguished from the system net as a p/t net (Lemma 3) and each p/t-like EOS is isomorphic to the system net as a p/t net (Lemma 4) .

EOS are a canonical extension of p/t nets, since the behaviour of an EOS when considering only the system net's perspective is in accordance with the behaviour of the system net considered as a p/t net, i.e. if a transition $\widehat{t}$ is disabled in the p/t net then for all ϑ the event $\widehat{t}[\vartheta]$ is disabled in the EOS.

Lemma 3. *For* $OS = (\widehat{N}, \mathcal{N}, d, \Theta, \mu_0)$ *define* $\Pi^1(OS) = \widehat{N}$*. For each* EOS OS *we have:*

$$\mu \xrightarrow[OS]{\widehat{t}[\vartheta]} \mu' \Longrightarrow \Pi^1(\mu) \xrightarrow[\Pi^1(OS)]{\widehat{t}} \Pi^1(\mu')$$

Proof. First, we have that $\Pi^1(\mu)$ is a marking of the p/t net $\widehat{N}$. Whenever μ enables $\widehat{t}[\vartheta]$ for a mode (λ, ρ) then $\phi(\widehat{t}[\vartheta], \lambda, \rho)$ holds which implies $\Pi^1(\lambda) = \mathbf{pre}(\widehat{t})$ and $\Pi^1(\rho) = \mathbf{post}(\widehat{t})$ and $\mu' = \mu - \lambda + \rho$.

Since $\mu \geq \lambda$ we have $\Pi^1(\lambda) \geq \Pi^1(\lambda) = \mathbf{pre}(\widehat{t})$, i.e. $\widehat{t}$ is enabled in $\Pi^1(\lambda)$.

For the system net projection follows:

$$\Pi^1(\mu') = \Pi^1(\mu - \lambda + \rho) = \Pi^1(\mu) - \Pi^1(\lambda) + \Pi^1(\rho) = \Pi^1(\mu) - \mathbf{pre}(\widehat{t}) + \mathbf{post}(\widehat{t})$$

This is the successor marking when firing $\widehat{t}$ in $\Pi^1(\mu)$ for the p/t net $\widehat{N}$. □

For a p/t-like EOS we have no object nets: $\mathcal{N} \setminus \{\bullet\} = \emptyset$, synchronisation given as $\Theta = \{\widehat{t}[\emptyset] \mid \widehat{t} \in \widehat{T}\}$, and the typing is the constant function $d = \bullet$ with $\bullet(\widehat{p}) = \bullet$ for all $\widehat{p} \in \widehat{P}$. The initial marking contains no submarking: $\mu_0 \in \widehat{P} \times \{\mathbf{0}\} \subseteq \mathcal{M}$. So, p/t-like EOS have the form:

$$OS = (\widehat{N}, \emptyset, \bullet, \{\widehat{t}[\emptyset] \mid \widehat{t} \in \widehat{T}\}, \mu_0)$$

Lemma 4. *A p/t-like* EOS $OS = (\widehat{N}, \emptyset, \bullet, \Theta_l, \mu_0)$ *is isomorphic to the p/t net* $(\widehat{N}, \Pi^1(\mu_0))$ *in the following sense:*

$$\mu \xrightarrow[OS]{\widehat{\tau}[\emptyset](\lambda,\rho)} \mu' \iff \Pi^1(\mu) \xrightarrow[\widehat{N}]{\widehat{\tau}} \Pi^1(\mu')$$

Proof. For a p/t-like EOS the predicate $\phi(\widehat{\tau}[\vartheta], \lambda, \rho)$ reduces to $\Pi^1(\lambda) = \mathbf{pre}(\widehat{\tau}) \wedge \Pi^1(\rho) = \mathbf{post}(\widehat{\tau})$ since $\mathcal{N} \setminus \{\bullet\} = \emptyset$. Therefore $\Pi^2(\mu) = \mathbf{0}$ holds for all reachable markings μ.

Since $\lambda \sqsubseteq \mu$ we have $\Pi^1(\lambda) \sqsubseteq \Pi^1(\mu)$, where $\Pi^1(\mu)$ is the marking in the p/t net $\widehat{N}$. The successor marking when firing $\widehat{\tau}[\emptyset](\lambda, \rho)$ in OS is defined as $\mu' = \mu - \lambda + \rho$. Obviously, $\Pi^2_N(\mu') = \mathbf{0}$ and $\Pi^1(\mu') = \Pi^1(\mu) - \mathbf{pre}(\widehat{\tau}) + \mathbf{post}(\widehat{\tau})$ which equals the successor marking when firing $\widehat{t}$ in $\widehat{N}$. □

4 A short Overview of related Nets-within-Nets Formalisms

The idea to use Petri nets as tokens – also called the "nets-within-nets" approach – can be traced back to the early nineties: Valk [1991] studies systems, where the net-tokens model the partial order of working plans that are executed within some external environment modelled as a Petri net again.

These nets are extended to elementary object net systems (EONS) in [Valk, 1998]. EONS are studied with respect to reference semantics, value semantics, and copy semantics (cf. Valk, 2003 for a more recent overview). Reference semantics is equivalent to our construction of the *reference net* $\text{RN}(OS)$. The value semantics of EONS defined in [Valk, 1998] is defined for the special cases of unary EOS and the class of generalised state machines.

Farwer [1999] studies the relationship of elementary object nets and Linear Logic.

There is also some connection to recursive Petri nets (RPN) [Haddad and Poitrenaud, 1999], where the firing of transitions can generate sub-net activity recursively. This nested threads of activity look somehow similar with a nesting of markings. The most obvious difference between RPN and EOS is the fact that the reachability problem is decidable for RPN (cf. Theorem 17 in Haddad and Poitrenaud, 1999) but undecidable for EOS (cf. Theorem 7).

Another variant of nets-within-nets is the formalism of PN^2 [Hiraishi, 2002] which allows to have several object nets within one system net and is mainly the same as EOS with a copy semantics.

Mobile Systems [Lakos, 2005] introduce another extension to object nets: modules that can interact via place and transition fusion. Modules describe locations and locations may be nested. Sub modules may shift from one module to another.

There are several formalism that extend the elementary case to systems with unbounded nesting. A first extension – called object nets – is defined in [Köhler and Rölke, 2004]. It can be shown that for an appropriate extension of the GSM property value and reference semantics are equivalent, too [Köhler and Rölke, 2005]. It is shown in [Köhler and Rölke, 2004] that object nets have the power to simulate Turing machines.

A very interesting restriction comes from the area of workflow nets: Adaptive workflow nets (AWFN) [Lomazova et al., 2006] restrict themselves to GSM and the net-tokens to workflow nets. The formalism is extended by the possibility to combine net-tokens at firing time with the usual workflow operations, like sequential composition, and-forks and or-decisions. Due to its restricted structure this formalism has some nice decidability properties.

Object nets are restricted in the sense that different levels of the system may synchronise, but cannot exchange markings directly. In [Köhler-Bußmeier and Heitmann, 2009] we defined the general case, i.e. object nets extended with communication channels, are defined. Of course, this extension cannot extend the expressibility any further, i.e. beyond that of Turing machines, but the expressiveness remains stable even if we restrict the net-tokens to carry at most one token on each place.

Similarly to object nets with communication channels, [Bednarczyk et al., 2004] defines object nets that allow to exchange object nets over communication channels, but they are restricted to the GSM case.

Reference nets [Kummer, 2002] are the generalisation of Eos for the case of arbitrary pointer structures. The semantics is based on graph rewriting, while Eos use term rewriting. Reference nets are supported by a very popular tool, called Renew [Kummer et al., 2004].

Another extension considers coloured tokens: Nested nets [Lomazova and Schnoebelen, 2000] can be seen as the extension of object nets in the direction of Coloured Petri Nets, i.e. we have tokens that are nets and tokens that are integers etc. Since Nested Nets have the possibility to destroy tokens it is clear that Nested Nets can simulate reset nets.

Another example for coloured formalisms are mobile predicate/transition nets [Xu and Deng, 2000] which are object nets under reference semantics and predicates as tokens.

AHO systems [Hoffmann et al., 2005] allow very complex data types as tokens which can be used to encode net-tokens as data types. This encoding works nicely for two levels but it seems that this cannot extended without further some clever, indirect coding.

An interesting extension of objects nets – discussed in [Köhler-Bußmeier, 2009] – allows algebraic operations on the net-tokens. This formalism therefore subsumes Adaptive workflow nets [Lomazova et al., 2006] as well as object nets with communication channels [Köhler-Bußmeier and Heitmann, 2009].

The formalism of minimal object-based nets (MOB nets) of Kummer [2000] is related to Eos but with quite different basic assumptions: MOB nets do not make any assumptions about the structure of the tokens; tokens just have a unique identity which can be compared with other identities and MOB nets can generate new tokens having fresh identifiers. It is shown in [Kummer, 2000] that these minimal assumptions are sufficient to show that reachability is undecidable for MOB nets – and therefore for every formalism dealing with name creation. The formalism of ν-abstract Petri nets [Velardo and de Frutos-Escrig, 2008] is quite similar to MOB nets. They have the same ability to create fresh names and have therefore also an undecidable reachability problem.

5 The Invariance Calculus for EOS

There is a well elaborated connection of Petri nets and linear algebra (cf. Lautenbach, 1987, Silva et al., 1998). In the following we recall basic definitions and properties for the case of p/t nets which we will then extend to Eos. Let $\Delta : T \to (P \to \mathbb{Z})$ be the function defined by:

$$\Delta(t)(p) = \mathbf{post}(t)(p) - \mathbf{pre}(t)(p)$$

$\Delta(t)$ denotes the *effect* of firing t. (Note that $\Delta(t)$ expresses less information than $\mathbf{pre}(t)$ and $\mathbf{post}(t)$. Only for nets without self loops both are equivalent.)

The function Δ is linear, in the sense that the effect $\Delta(t_1 + t_2)$ of a transition multiset is their cumulated effect:

$$\Delta(t_1 + t_2) = \Delta(t_1) + \Delta(t_2)$$

If $0 < |P|, |T| < \infty$ then Δ can be expressed as a $|P| \cdot |T|$ matrix (called *incidence matrix*) defined by $\Delta(p,t) = \mathbf{post}(t)(p) - \mathbf{pre}(t)(p)$. Using $\Delta(t)$ the successor marking of the firing step $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ in a p/t net can be calculated as:

$$\mathbf{m}' = \mathbf{m} - \mathbf{pre}(t) + \mathbf{post}(t) = \mathbf{m} + \Delta(t)$$

It is well known that all solutions $\mathbf{i} \in \mathbb{Z}^{|P|} \setminus \{\mathbf{0}\}$ of the equation

$$\Delta^\top \mathbf{i} = \mathbf{0}$$

which are called place-invariants (short: P-invariants) result into a linear equation that holds for all reachable markings $\mathbf{m} \in RS(N, \mathbf{m}_0)$. The following captures the essence of place-invariants.

Theorem 2 (Lautenbach). *Let $\mathbf{i} \in \mathbb{Z}^{|P|}$ be a P-invariant of the p/t net N. Then we have:*

$$\forall \mathbf{m} \in RS(N, \mathbf{m}_0) : \mathbf{i} \cdot \mathbf{m} = \mathbf{i} \cdot \mathbf{m}_0$$

This invariance calculus for p/t nets can be extended to Eos in a compositional way, i.e. invariance equations can be obtained from the invariance equations of the constituting components separately.

Theorem 3. *Let $OS = (\widehat{N}, \mathcal{N}, d, l, \mu_0)$ be an* Eos, *$\widehat{\mathbf{i}}$ a P-invariant of the system net $\widehat{N}$ and $\mathbf{i}_N$ one for each object net $N \in \mathcal{N}$. For all reachable markings μ it holds:*

$$\begin{aligned} \widehat{\mathbf{i}} \cdot \Pi^1(\mu) &= \widehat{\mathbf{i}} \cdot \Pi^1(\mu_0) \\ \forall N \in \mathcal{N} : \mathbf{i}_N \cdot \Pi^2_N(\mu) &= \mathbf{i}_N \cdot \Pi^2_N(\mu_0) \end{aligned}$$

Proof. Proof by induction on the length of the firing sequence. Induction base: For the empty sequence we have $\mu = \mu_0$ and the property is obvious.

Induction step: Assume we have $\mu_0 \xrightarrow[OS]{*} \mu \xrightarrow[OS]{\widehat{t}[\vartheta](\lambda,\rho)} \mu'$. Since $\widehat{\mathbf{i}}$ is an invariant of the system net we have $\widehat{\mathbf{i}} \cdot (\mathbf{post} - \mathbf{pre}) = \mathbf{0}$. It follows:

$$\begin{aligned} \widehat{\mathbf{i}} \cdot \Pi^1(\mu') &= \widehat{\mathbf{i}} \cdot \Pi^1(\mu - \lambda + \rho) = \widehat{\mathbf{i}} \cdot (\Pi^1(\mu) - \Pi^1(\lambda) + \Pi^1(\rho)) \\ &= \widehat{\mathbf{i}} \cdot \Pi^1(\mu) - \widehat{\mathbf{i}} \cdot \mathbf{pre}(\widehat{t}) + \widehat{\mathbf{i}} \cdot \mathbf{post}(\widehat{t}) = \widehat{\mathbf{i}} \cdot \Pi^1(\mu) \end{aligned}$$

For all $N \in \mathcal{N}$ we have $\mathbf{i}_N \cdot (\mathbf{post}_N - \mathbf{pre}_N) = \mathbf{0}$. It follows:

$$\mathbf{i}_N \cdot \Pi^2_N(\rho) = \mathbf{i}_N \cdot \big(\Pi^2_N(\lambda) - \mathbf{pre}_N(\vartheta(N)) + \mathbf{post}_N(\vartheta(N))\big) = \mathbf{i}_N \cdot \Pi^2_N(\lambda)$$

This proves the property. □

This compositionality property reduces the complexity to compute invariants in a substantial way. This can be seen by considering the special case, where we assume that the EOS is a generalised state machine. In this case one obtains all invariants of the EOS OS by computing those of the reference net $\text{RN}(OS)$ which is a p/t net. If the size of all nets, i.e. the system net and all object nets, is less than n, than the size of the reference net $\text{RN}(OS)$ is roughly $(|\mathcal{N}| + 1) \cdot n$. If the number of nets $|\mathcal{N}|$ and their sizes n are of comparable order this means a quadratic increase in computational resources.

This extension of linear invariants to EOS shows that safety properties of object nets – when considered as p/t nets – are conservatively embedded. Of course, this embedding does not extend to liveness properties, since e.g. a deadlock-free p/t net may block when used as a system net of an EOS, simply because it may be synchronised with a deadlocked object net.

Example 4. As mentioned in the introduction structural analysis is useful for the system's as well as for the mobile agent's side. The mobility infrastructure given in Fig. 4 consists of the three localities *pool*, *public*, and *private*. The net itself is a variant of the reader/writer problem. The parameter $n \in \mathbb{N}$ denotes the capacity of the public location.

In the first step only the agent system (i.e. the system net $\widehat{N}$) is shown, since an agent (i.e. the object net N) cannot be restricted by a platform in advance. The synchronisation relation is also omitted for the same reason.

We show how invariants of the system net extend towards properties of the whole EOS. The following analysis holds for arbitrarily structured agents.

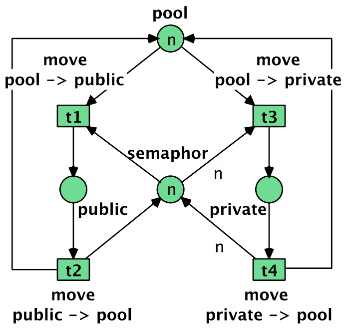
There are three locations: *pool*, *public*, and *private*. The pool location is the initialisation area; the public area is open for any agent, while the private area has restricted access: It is allowed that many agents are simultaneously in the public location, but there can be at most one agent in the private location. This prevents agents from being spied out. The transitions between the locations model movement, which are either objective or consensual (depending on the synchronisation relation).

In the following the system net $\widehat{N}$ is analysed using invariants. We obtain $\widehat{\mathbf{i}} = (0, 1, 1, n)'$ as a solution of the equation $\widehat{\mathbf{i}} \cdot \Delta = \mathbf{0}$. Using Theorem 3 we have $\widehat{\mathbf{i}} \cdot \Pi^1(\mu) = \widehat{\mathbf{i}} \cdot \Pi^1(\mu_0)$ for all reachable markings μ:

$$\widehat{\mathbf{i}}_1 \cdot \Pi^1(\mu) = \Pi^1(\mu)(\mathit{public}) + \Pi^1(\mu)(\mathit{semaphor}) + n \cdot \Pi^1(\mu)(\mathit{private}) = \widehat{\mathbf{i}}_1 \cdot \Pi^1(\mu_0) = n$$

Therefore $\Pi^1(\mu)(\mathit{private}) > 0$ implies $\Pi^1(\mu)(\mathit{private}) = 1$ and $\Pi^1(\mu)(\mathit{public}) = 0$.

In the following we analyse the agent net $\widehat{N}$ given in Fig. 5. The two places flag_1 and flag_2 are used to toggle the agent's choice between the public and the



$\Delta =$

	t_1	t_2	t_3	t_4
pool	-1	1	-1	1
public	1	-1		
semaphor	-1	1	$-n$	n
private			1	-1

Fig. 4. The Multi-Agent System Net $\widehat{N}$

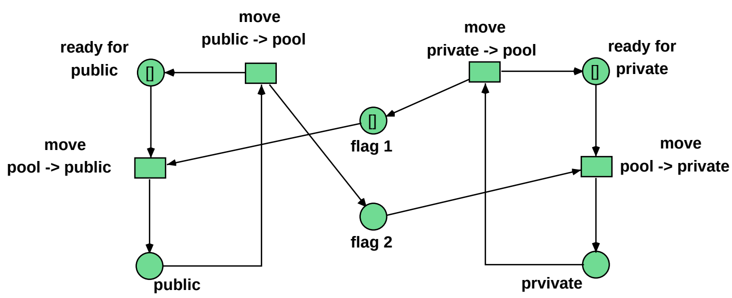


Fig. 5. The Agent

private place. The incidence matrix is given as (with slight abbreviations):

$\Delta =$

	pool $\to$ *pub*	*pub* $\to$ *pool*	*pool* $\to$ *prv*	*prv* $\to$ *pool*
ready public	-1	1		
public	1	-1		
ready private			-1	1
private			1	-1
flag$_1$	-1			1
flag$_2$		1	-1	

Solving the equation $\mathbf{i} \cdot \Delta = \mathbf{0}$ we obtain $\mathbf{i} = (0, 1, 0, 1, 1, 1)'$ as an invariant of the agent-net. Using Theorem 3 we have $\mathbf{i} \cdot \Pi^2(\mu) = \mathbf{i} \cdot \Pi^2(\mu_0)$ for all reachable markings μ:

$$\begin{aligned}\mathbf{i} \cdot \Pi^2(\mu) &= \Pi^2(\mu)(public) + \Pi^2(\mu)(private) + \Pi^2(\mu)(flag_1) + \Pi^2(\mu)(flag_2) \\ &= \mathbf{i} \cdot \Pi^2(\mu_0) = 1\end{aligned}$$

This implies:

$$\Pi^2(\mu)(public) + \Pi^2(\mu)(private) \leq 1$$

So, we have proven that the agent does not attempt to enter the private and the public place at the same time.

6 Decidability Problems for EOS

The interesting part in the firing rule of Eos is the fact that moving an object net-token in the system net has the power to modify the state of an unbounded number of tokens, i.e. all the tokens of the object net-tokens (including the case of zero tokens). It is therefore a natural question whether this increases the expressiveness of Eos compared to p/t nets. Here we consider the most well known decidability problems for Petri nets: The reachability, the liveness and the boundedness problem.

For the reachability problem one has to decide whether $m_1 \xrightarrow{*} m_2$ for a given p/t net N and two markings m_1 and m_2. Reachability has been studied for p/t nets and for variants of object nets: The reachability problem for p/t nets is studied in [Araki and Kasami, 1977].

For the liveness problem one has to decide whether all transitions of a given p/t net N and its initial marking m_0 are live. A transition t is live if for all markings m reachable from m_0 there exists one marking m' reachable from m that enables t:

$$\forall m \in RS(m_0) : \exists m' \in RS(m) : m' \xrightarrow{t}$$

Since it is known for a long time that reachability and liveness are equivalent problems (cf. Theorem 1.6 and 1.9 in Jantzen and Valk, 1980 or Theorem 5.5 and 5.6 in Peterson, 1981) the question whether both are decidable or not was open for several years. Decidability of reachability for p/t nets is shown in [Mayr, 1984], a different proof is given in [Lambert, 1992].

Boundedness is the problem to decide whether there are only finitely many reachable markings. The boundedness problem is decidable for p/t nets [Karp and Miller, 1969] which is due to the fact that p/t nets enjoy *monotonicity*: If transition t is enabled in marking m_1 then it is also enabled for each greater marking m_2. Formally:

$$\forall m_1, m_1', m_2 : (m_1 \xrightarrow{t} m_1' \wedge m_1 < m_2) \Longrightarrow (\exists m_2' : m_2 \xrightarrow{t} m_2' \wedge m_1' < m_2')$$

Here $<$ denotes the strict order on multisets. This fact leads to the construct of a coverability graph which is always finite and which is expressive enough to identify the unbounded places. Then boundedness is decidable for some extensions of Petri nets – like Post-SM Nets and Transfer Nets – and undecidable for Reset Nets, Inhibitor Nets, and Self-Modifying Nets (cf. Dufourd et al., 1998 for details).

6.1 Simulation of Counter Programs

It is well-known that it is undecidable whether a counter program with at least two counters will terminate. In the following we show how to simulate counter programs by Eos.

Definition 4. *A* counter program *CP using m counters $c_1, \ldots, c_m$ is a finite sequence of commands ending with the halt statement:*

$$CP = cmd_1; \ldots; cmd_{n-1};\ \mathsf{halt}$$

There are three types of commands ($1 \leq j \leq m$ and $1 \leq k_1, k_2 \leq n$):

$$c_j := c_j + 1, \quad c_j := c_j - 1, \quad or \quad \mathsf{ifzero}\ c_j\ \mathsf{jump}\ k_1\ \mathsf{else}\ k_2$$

A configuration *$C = (k, n_1, \ldots, n_m)$ denotes the values n_j of the counters c_j and the current position k in the program. The initial configuration is $C_0 = (1, 0, \ldots, 0)$.*

The successor configuration *$C' = (k', n_1', \ldots, n_m')$ of $C = (k, n_1, \ldots, n_m)$ depends on the current command cmd_k:*

- *$c_j := c_j + 1$. Then $k' = k + 1$, $n_j' = n_j + 1$ and $n_i' = n_i$ for all $i \neq j$.*
- *$c_j := c_j - 1$. Whenever $n_j > 0$ holds, then $k' = k + 1$, $n_j' = n_j - 1$ and $n_i' = n_i$ for all $i \neq j$. For $n_j = 0$ an error occurred and the program blocks.*
- *$\mathsf{ifzero}\ c_j\ \mathsf{jump}\ k_1\ \mathsf{else}\ k_2$. Then $n_i' = n_i$ for all i and $k' = k_1$ if $n_j = 0$ and $k' = k_2$ otherwise.*
- *halt terminates the execution successfully.*

The successor configuration C' of C (whenever defined) is uniquely determined. The change on configuration is denoted $C \xrightarrow[CP]{} C'$.

For Eos with unrestricted typing d it is possible to give a bisimulation of counterprograms. Figure 6 shows the direct translation of a counter program statement cmd_k into a net fragment $\widehat{N}(cmd_k)$. The fragments share the common counter places cnt_j for all $j \in \{1, \dots, m\}$ and the state places q_k for all $k \in \{1, \dots, n\}$.

The fragments are part of a system net which has net-tokens of the object net type N_c (also shown in Figure 6). This object net has only one place *counter* and three transitions which are used for incrementation, decrementation, and the test for non-emptiness.

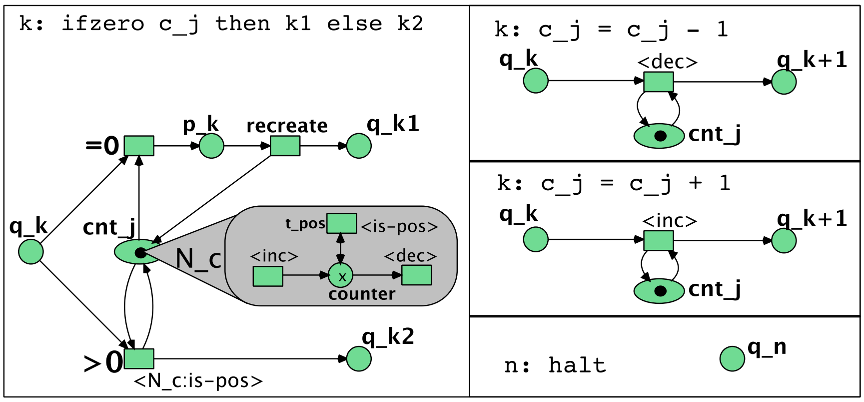


Fig. 6. The Eos-translation of counter commands

Definition 5. *Let* $CP = cmd_1; \dots; cmd_{n-1};$ halt *be a* m*-counter program. The simulating* Eos *is defined as*

$$OS_{strong}(CP) = (\widehat{N}, \mathcal{N}, d, l, \mu_0) \quad with \quad \mathcal{N} = \{\bullet, N_c\}$$

- *The system net* $\widehat{N}$ *is defined as the net-union of all net fragments* $\widehat{N}(cmd_k)$ *for* $k \in \{1, \dots, n\}$*. as given in Figure 6.*
- *The object net* N_c *is given as in Figure 6.*
- *The synchronisation* l *is given by the labels* inc*,* dec *and* is-pos*.*
- *The type* $d(p)$ *of the system net places is* $d(cnt_j) = N_c$ *for* $j \in \{1, \dots, m\}$ *and* $\bullet$ *otherwise.*
- *The initial marking is* $\mu_0 = q_1[] + \sum_{j=1}^{m} cnt_j[\mathbf{0}]$.

The marking μ of the Eos $OS_{strong}(CP)$ encodes the configuration $C = (k, n_1, \dots, n_m)$, denoted $\mu = \mu(C)$, if the place q_k is marked with a black token and for each counter c_j the counter place cnt_j is marked with one net-token that has n_j tokens on the place *counter*. Formally:

$$\mu((k, n_1, \dots, n_m)) := q_k[] + \sum_{j=1}^{m} cnt_j[n_j \cdot counter]$$

We now show how one step of the counter automaton and the test for zero is simulated by the Eos $OS_{strong}(CP)$. For each cmd_k there is a firing sequence $\sigma(cmd_k)$ in $OS_{strong}(CP)$. Additionally, we have that for each initial marking the Eos $OS_{strong}(CP)$ has exactly one maximal firing sequence (which can either be finite or infinite), i.e. there is no concurrency and no non-deterministic conflict present.

Lemma 5. *Each computation of CP is bisimulated by a firing sequence of* $OS_{strong}(CP)$*. Each command cmd is simulated by a sequence* $\sigma(cmd)$ *in the following sense:*

$$C \xrightarrow[CP]{cmd} C' \iff \mu(C) \xrightarrow[OS_{strong}(CP)]{\sigma(cmd)} \mu(C')$$

Additionally, whenever configuration C *executes cmd as its next command, then* $\mu(C)$ *has* $\sigma(cmd)$ *as the only (maximal) enabled firing sequence.*

Proof. The basic idea is to model the counter c_j by an object net on the place cnt_j (cf. Figure 6) and its value by n_j tokens on the place *counter*. Incrementation and decrementation are directly simulated via one synchronised event. The test whether the counter is greater than zero is realised by a synchronisation of the system net transition >0 with the object net transition is-pos. In these cases the firing sequence $\sigma(cmd)$ has length one.

But also the test for $c_j = 0$ is simulated directly: The transition =0 is enabled if and only if the counter place cnt_j is marked with a net-token with an empty marking (i.e. the encoding of $c_j = 0$), since the transition =0 has the object net N_c in the preset but not in the postset (i.e. the typing is not conservative). As we have seen above the firing rule enforces all net-tokens of type N_c to be unmarked. The transition recreate generates a new object net with an empty marking (as required) on the place cnt_j. Here $\sigma(cmd)$ has length two.

Thus, the system net transition >0 is enabled if and only if the counter is positive and the system net transition =0 is enabled if and only if the counter is zero. □

Due to this strong simulation we obtain the following undecidabilty results.

Theorem 4. *Reachability, boundedness, and coverability are undecidable for* Eos.

Proof. These problems are undecidable for inhibitor nets (cf. Dufourd et al., 1998) and each inhibitor net is bisimulated by a counterprogram which is bisimulated by the (non-conservative) Eos $OS_{strong}(CP)$. □

6.2 Simulation of Inhibitor Nets

The proof of Theorem 4 relies on a reduction of counterprograms to Eos. The same result can be obtained by giving a direct simulation of inhibitor nets. It is well-known that the reachability problem is undecidable for inhibitor nets. In the following we show how to simulate inhibitor nets by Eos.

Lemma 6. *For each inhibitor net N^* there is an* Eos $OS_{strong}(N^*)$ *that simulate* N^*.

Proof. We show that each inhibitor net can be simulated by the Eos $OS_{strong}(N^*)$. Without loss of generality we consider inhibitor nets without arc weights and we assume that for each transition we have that whenever a place p is connected via a inhibitor arc then this place is not connected with t via a normal arc. Let us consider a inhibitor net given as $N^* = (P^*, T^*, F^*, F_r^*, m_0)$, where $F_r^* \subseteq P^* \times T^*$ describes the inhibitor arcs. A transition t is enabled in m iff there is at least one token on each input place and all inhibitor place carry the empty marking, i.e. $m(p) \geq F(p,t)$ for all p and $m(p) = 0$ for all p such that $(p,t) \in F_{inh}^*$.

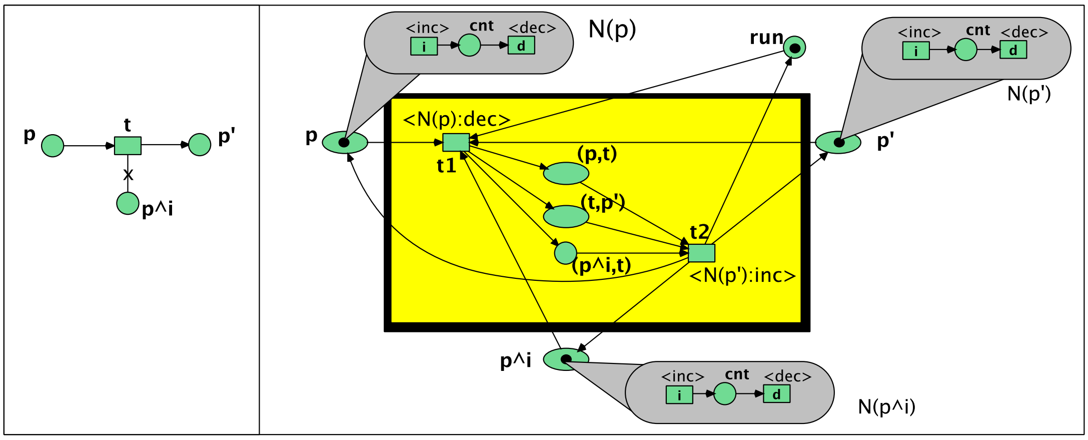


Fig. 7. The Eos-translation of inhibitor nets

Each marking m of the inhibitor net is encoded as the marking $\mu(m)$ of the Eos. We say that a nested marking μ encodes a marking m of N^* whenever μ contains exactly one net-token on each place $p \in P^*$ (and none on the other places) and the net-token on p has exactly $m(p)$ tokens on its place cnt:

$$\mu(m) := \mathsf{run}[] + \sum_{p \in P^*} p[m(p) \cdot \mathsf{cnt}_{d(p)}]$$

Each firing $m \xrightarrow{t} m'$ is simulated deterministically by the firing $\mu(m) \xrightarrow{t_1 t_2} \mu(m')$.

The simulating Eos $OS_{strong}(N^*) = (\widehat{N}, \mathcal{N}, d, \Theta, \mu_0)$ is constructed in the following way:

- For each place $p \in P^*$ in the inhibitor net the simulating Eos has one object-net $N(p)$. Each object-net $N(p)$ has exactly one place $\mathsf{cnt}_{N(p)}$ and the two transitions $\mathsf{i}_{N(p)}$ and $\mathsf{d}_{N(p)}$, where $\mathsf{i}_{N(p)}$ is labelled with channel $\mathsf{inc}_{N(p)}$ and $\mathsf{d}_{N(p)}$ is labelled with channel $\mathsf{dec}_{N(p)}$. In particular all the object nets $N(p)$ have the same net structure. Additionally we have the object-net $\bullet$:

$$\mathcal{N} = \{\bullet\} \cup \{N(p) \mid p \in P^*\}$$

- The system net $\widehat{N}$ is obtained from the inhibitor N^* via a substitution for each transition which is illustrated in Figure 7:
  Each transition $t \in T^*$ is replaced by two transitions t_1 and t_2.

$$\widehat{T} := \{t_1, t_2 \mid t \in T^*\}$$

  For each input arc $(p,t) \in F^* \cap (P^* \times T^*)$ we add the place (p,t); for each output arc $(t,p') \in F^* \cap (T^* \times P^*)$ we add the place (t,p'); for each inhibitor arc $(p^i,t) \in F_r^*$ we add the place (p^i,t). Additionally, we have one global *run* place which guarantees that firing of t_1 must be followed by t_2 before any other transition can fire.

$$\widehat{P} := P^* \cup F^* \cup F_r^* \cup \{\mathsf{run}\}$$

  For each input arc (p,t) the transition t_1 is labelled with $\mathsf{dec}_{N(p)}$:

$$\widehat{l}(t_1)(N(p)) = \begin{cases} \mathsf{dec}_{N(p)}, & \text{if } (p,t) \in F^* \cap (P^* \times T^*) \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

  For each output arc (t,p) the transition t_2 is labelled with $\mathsf{inc}_{N(p')}$:

$$\widehat{l}(t_2)(N(p')) = \begin{cases} \mathsf{inc}_{N(p')}, & \text{if } (t,p') \in F^* \cap (T^* \times P^*) \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

- The typing d is defined as:

$$d(p) = N(p) \qquad d(p,t) = N(p) \qquad d(t,p') = N(p') \qquad d(p^i,t) = d(\mathsf{run}) = \bullet$$

- The initial marking is defined as the encoding of m, i.e. $\mu_0 := \mu(m_0)$.

Whenever a place p^i is connected via a inhibitor arc with t then t_1 has exactly one place of type $N(p^i)$ in its preset but none in the postset. Therefore t_1 can only fire if the marking of the net-token is the empty multiset. Whenever t_2 fires it generates one net-token on p^i again which must be empty since there is no place of type $N(p^i)$ in the preset of t_2. It is straightforward to see that we have:

$$m \xrightarrow{t} m' \iff \mu(m) \xrightarrow{t_1 t_2} \mu(m')$$

This proves that the Eos $OS_{strong}(N^*)$ simulates the inhibitor net N^*. □

We define the liveness problem for Eos analogously to that of p/t nets: For the liveness problem one has to decide whether all events $\theta \in \Theta$ of a given Eos OS are live. An event θ is live if for all markings μ reachable from μ_0 there exists a marking μ' reachable from μ that enables θ.

Lemma 7. *The reachability problem is reducible to the liveness problem for* Eos.

Proof. The proof follows the idea given in [Peterson, 1981] that shows the equivalence of reachability and liveness for p/t nets.

It is sufficient to consider the problem whether the empty marking is reachable since for each inhibitor net N_1 and each marking m we can construct another inhibitor net N_2 with the property: The marking m is reachable in N_1 iff $\mathbf{0}$ is reachable in N_2. The net N_2 is obtained from N_1 by adding one place run and one transition t (cf. Fig. 8). The additional run-place is attached as a side condition to each transition of N_1. Initially the place run is marked with one token. The additional transition t removes exactly $m(p)$ tokens from each p (where m is the given marking tested for reachability) and one token from run. The postset of t is empty. It is obvious that N_2 has the desired property.

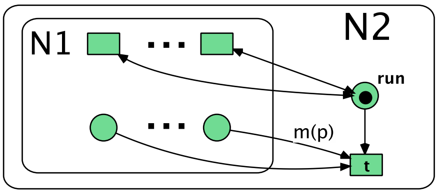


Fig. 8. Reduction from Reachability to Reachability of the Empty Marking

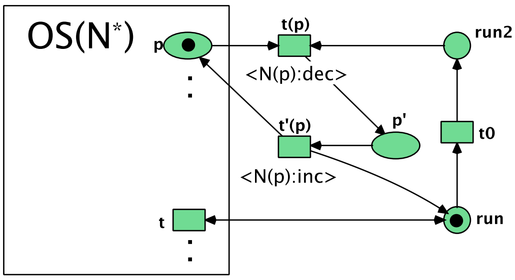


Fig. 9. Reduction from the **0**-Reachability to Liveness of t_0

We will construct an Eos $OS(N^*)$ from a given p/t net N^* such that the empty marking is reachable in N^* iff the event $t_0[\vartheta]$ is not live in $OS(N^*)$.

Assume the inhibitor net is given as $N^* = (P^*, T^*, F^*, F^*_{inh}, m_0)$. We define $OS(N^*)$ almost the same as in Lemma 6. We add transition t_0 and the place run_2 and for each $p \in P^*$ the place p' and the transitions $t(p)$ and $t'(p)$ (cf. Fig. 9). We set $d(\mathsf{run}_2) := \bullet$ and $d(p') := d(p) = N(p)$. Remark: Since t_0 has only places of the black token type in the pre- and postset (i.e. $\mathcal{N}(t_0) = \{\bullet\}$) we obtain that if the event $t_0[\vartheta]$ is activated then ϑ is uniquely determined as $\vartheta(N) = \mathbf{0}$ holds for all $N \in \mathcal{N}(t_0)$.

As before, we define $\mu(m)$ as:

$$\mu(m) := \mathsf{run}[] + \sum_{p \in P^*} p[m(p) \cdot \mathsf{cnt}_{d(p)}]$$

A marking that is reachable in N^* is so in $OS(N^*)$:

$$m_1 \xrightarrow{w} m_2 \Longrightarrow \mu(m_1) \xrightarrow{w} \mu(m_2)$$

Assume that $\mathbf{0}$ is reachable in N^*. In $\mu(\mathbf{0})$ we have $\mu(\mathbf{0}) \xrightarrow{t_0[\vartheta]} \mu := \mathsf{run}_2[] + \sum_{p \in P^*} p[\mathbf{0}]$ and in μ no event is activated anymore. So, if $\mathbf{0}$ is reachable in N^* then clearly $t_0[\vartheta]$ is not live.

Assume that $\mathbf{0}$ is not reachable in N^*. Then $t_0[\vartheta]$ is live: For each marking $m^* \neq \mathbf{0}$ we have $m^*(p_0) > 0$ for some p_0 and therefore we have $\mu(m^*) \xrightarrow{t_0[\vartheta]} \mu' \xrightarrow{t(p_0)[\vartheta']t'(p_0)[\vartheta'']} \mu(m^*)$. Note that $t(p_0)[\vartheta']t'(p_0)[\vartheta'']$ does not alter the marking of the net-token on p. □

We can also study the variant of *group liveness*, where we try to activate $t[\vartheta]$ not for a fixed, but only for some ϑ: A transition t is *group-live* if for all markings μ reachable from μ_0 there exists a marking μ' reachable from μ that enables $t[\vartheta]$ for some ϑ. From the proof we can see that also group-liveness is undecidable since the empty marking is reachable in N^* iff t_0 is group-live in $OS(N^*)$.

Theorem 5. *Reachability, liveness, boundedness, and coverability are undecidable for* Eos.

7 Decidability Results for Conservative EOS

The expressiveness of Eos as formulated in Theorem 4 was due to a non-conservative typing. Recall, that a typing d is called conservative if for all t we have that each place $\widehat{p}$ in its preset there is a place $\widehat{p}'$ in its postset typed with the same net, i.e. $d(\widehat{p}) = d(\widehat{p}')$.

If this condition is violated, i.e. we have a system net transition $\widehat{t}$ such that $N \in d({}^\bullet\widehat{t})$ and $N \notin d(\widehat{t}^\bullet)$ for some object net N, then firing of this transition enforces net-tokens of type N to be unmarked (emptiness constraint): The event $\widehat{\tau}[\vartheta]$ is enabled in mode (λ, ρ) only if all object nets in λ of this type N carry the empty marking: $\Pi^2_N(\lambda) = \mathbf{0}$ (cf. the definition of the enabling predicate ϕ in (6)). In other words: The system net cannot destroy the tokens within a net-token.[6] It is exactly this blocking behaviour that we used in the proof of Theorem 4 to simulate counter programs. Note that blocking destroys the monotonicity of the firing rule.

[6] There are variants of the formalism that allow the destruction of the net-token's tokens (i.e. the copy semantics discussed in Valk, 2003). But then it is quite obvious that one can simulate at least reset nets i.e. Petri nets with reset arcs. It is known that reachability and boundedness is undecidable for reset nets while e.g. coverability remains decidable (cf. Dufourd et al., 1998).

The symmetric case of typing is unproblematic: A transition $\widehat{t} \in \widehat{T}$ with an object net N that is present in the postset, but not in the preset, i.e. $N \notin d(^\bullet\widehat{t})$ and $N \in d(\widehat{t}^\bullet)$ generates net-tokens of type N. The firing rule ensures that these net-tokens carry the empty marking since in this case $\widehat{\tau}[\vartheta]$ is enabled in mode (λ, ρ) only if all object nets in ρ of this type N carry the empty marking.

7.1 Monotonicity of the Firing Rule for Conservative EOS

One can argue that the blocking is an definitorial artefact. Therefore we study Eos without blocking situations, i.e. conservative Eos. It will turn out that the blocking behaviour is somehow the only source of the equivalence of Eos to counter programs: If we consider Eos without this blocking behaviour, i.e. conservative Eos, we will regain the monotonicity property of the firing rule.

We define the order $\preceq$ on nested markings by:

$$\begin{aligned} \alpha \preceq \beta \qquad \iff & \alpha = \textstyle\sum_{i=1}^{m} \widehat{a}_i[A_i] \wedge \beta = \sum_{i=1}^{n} \widehat{b}_i[B_i] \wedge \\ & \forall 1 \leq i \leq m : \widehat{a}_i = \widehat{b}_i \wedge A_i \leq B_i \end{aligned} \tag{9}$$

Note that $\alpha \sqsubseteq \beta$ is a special case of $\alpha \preceq \beta$, where $A_i \leq B_i$ is restricted to $A_i = B_i$.

It is clear that for a transition $\widehat{t}$ such that $N \in d(^\bullet\widehat{t})$ and $N \notin d(\widehat{t}^\bullet)$ for some object net N, the enabling of $\widehat{\tau}[\vartheta]$ in the mode (λ, ρ) does *not* imply the enabling of $\widehat{\tau}[\vartheta]$ in all (λ', ρ') with $\lambda \preceq \lambda'$, i.e. the firing rule for general Eos is not monotonous.

Therefore we have to forbid typings, where $N \in d(^\bullet\widehat{t})$ and $N \notin d(\widehat{t}^\bullet)$ for some transition $\widehat{t}$, i.e. we restrict Eos to conservative ones.

Given the representation above for $\alpha \preceq \beta$, then $\sum_{i=1}^{m} \widehat{a}_i[B_i]$ is called a *α-restriction of β*. In general there are many restrictions since the sum representation of α as $\sum_{i=1}^{m} \widehat{a}_i, [A_i]$ and β as $\sum_{i=1}^{n} \widehat{b}_i[B_i]$ are not unique. Let $(\beta \downarrow \alpha)$ denote the set of all α-restrictions of β. Let α and β be arbitrary nested multisets with $\alpha \preceq \beta$. Then we have:

$$\forall \alpha, \beta \in \mathcal{M} : \alpha \preceq \beta \Longrightarrow \forall \gamma \in (\beta \downarrow \alpha) : \alpha \preceq \gamma \preceq \beta \wedge \Pi^1(\alpha) = \Pi^1(\gamma) \leq \Pi^1(\beta) \tag{10}$$

Lemma 8. *For* Eos *with conservative typing d the firing rule is monotonous w.r.t. the order $\preceq$, i.e. if the event $\widehat{\tau}[\vartheta]$ is enabled then it is enabled for each greater marking:*

$$(\forall \mu_1, \mu_1', \mu_2 : \mu_1 \xrightarrow[OS]{\widehat{\tau}[\vartheta]} \mu_1' \wedge \mu_1 \prec \mu_2) \Longrightarrow (\exists \mu_2' : \mu_2 \xrightarrow[OS]{\widehat{\tau}[\vartheta]} \mu_2' \wedge \mu_1' \prec \mu_2')$$

Proof. Let (λ_1, ρ_1) be the mode of $\mu_1 \xrightarrow[OS]{\widehat{\tau}[\vartheta]} \mu_1'$ and let $\mu_1 \prec \mu_2$. We construct a mode (λ_2, ρ_2) for μ_2. We choose a λ_1-restriction of μ_2 for λ_2, i.e. $\lambda_2 \in (\mu_2 \downarrow \lambda_1)$. By (10) this implies $\Pi^1(\lambda_1) = \mathbf{pre}(\widehat{\tau}) = \Pi^1(\lambda_2)$ and $\lambda_1 \preceq \lambda_2$ and we know that

λ_2 involves the same number of net-tokens for each place in the system net and all the net-tokens have a greater marking each, cf. (9).

Therefore the object nets' tokens $\Pi^2_N(\lambda_2) - \Pi^2_N(\lambda_1) \geq \mathbf{0}$ are irrelevant for ϕ. They can be added arbitrarily to the net-tokens in ρ_1 to construct ρ_2. All these choices are equivalent w.r.t. the projection equivalence $\cong$. For each possible ρ_2 the mode (λ_2, ρ_2) enables $\widehat{\tau}[\vartheta]$ in μ_2.

The successor marking μ_2' is greater than μ_1': We have $\mu_1 \geq \lambda_1$ by Def. 3 and by this $\mu_1 = \mu_{0,1} + \lambda_1$. The successor is $\mu_1' = \mu_{0,1} + \rho_1$. Analogously $\mu_2 = \mu_{0,2} + \lambda_2$ and $\mu_2' = \mu_{0,2} + \rho_2$. The nested multisets $\mu_{0,i}, i = 1, 2$ contain all the net-tokens which are untouched. Since $\mu_1 \prec \mu_2$ we know that one of the inclusion must be strict: $\mu_{0,1} \prec \mu_{0,2}$ or $\lambda_1 \prec \lambda_2$. Since $\rho_1 \preceq \rho_2$ holds in the first case and $\rho_1 \prec \rho_2$ in the second one we obtain $\mu_1' \prec \mu_2'$. □

In the following we show that the reachability graph of a conservative EOS is a well structured transitions system [Abdulla et al., 1996, Finkel and Schnoebelen, 2001].

A quasi order is a reflexive and transitive binary relation. A partial order is a antisymmetric quasi order. For a given partial order $\leq$ on the set A the upward closure of a subset B is $\uparrow B := \{x \in A \mid \exists b \in B : b \leq x\}$. For a given partial order $\leq$ on the set A the set of minimal elements for $B \subseteq A$ is a set $\min(B) \subseteq B$ of

A quasi order is a reflexive and transitive binary relation. A partial order is a antisymmetric quasi order. For a given partial order $\leq$ on the set A the upward closure of a subset B is $\uparrow B := \{x \in A \mid \exists b \in B : b \leq x\}$. For a given partial order $\leq$ on the set A the set of minimal elements for $B \subseteq A$ is a set $\min(B) \subseteq B$ of incomparable elements such that for all $b \in B$ there is an element $a \in \min(B)$ with $a \leq b$.

A quasi order $\leq$ over the set X is called *well quasi order* (wqo) whenever each sequence $x_0 x_1 x_2 \ldots$ contains two comparable elements, i.e. there are elements x_i and x_j with $i < j$ such that $x_i \leq x_j$. Therefore, any wqo prohibits sequences with infinite incomparable elements. In a wqo the set $\min(B)$ is always finite.

A transition system $(S, \to, S_0)$ consists of a set of states S, initial states $S_0 \subseteq S$ and a transition relation $\to \subseteq S \times S$. For $S' \subseteq S$ we denote the set of predecessors by $\mathit{Pred}(S') := \{s \in S \mid s \to s' \in S'\}$ and successors by $\mathit{Succ}(S') := \{s \in S \mid s' \to s, s' \in S'\}$. A transition system is finitely branching if $\mathit{Succ}(s)$ is finite for all states s.

A partial order $\leq$ on the set of states S of a transition system $(S, \to, S_0)$ has the property of strict compatibility with respect to $\leq$ (sometimes called monotonicity), if the following holds: Whenever $s_1 \to s_1'$ and $s_1 < s_2$, then there exists a s_2' such that $s_2 \to s_2'$ with $s_1' < s_2'$. Similarly, a transition system has the property of compatibility if the above holds for $\leq$ instead of $<$, i.e. for the non-strict case. In general, strict compatibility implies compatibility.

A *well structured transition system (wsts)* $(S, \to, S_0, \leq)$ is a finitely branching transitions system $(S, \to, S_0)$ with a strict compatible order $\leq$ such that $\leq$ is a decidable wqo and $\min(\mathit{Pred}(\uparrow s))$ is computable for all $s \in S$.

Theorem 6. *Boundedness and Coverability are decidable for conservative* Eos.

Proof. Generalising the result of [Karp and Miller, 1969] it is shown in [Finkel and Schnoebelen, 2001] that the boundedness and the coverability problem are decidable for well structured transitions system. The reachability graph of a conservative Eos is a well structured transitions system: By Lemma 8 the transition system is strictly compatible with the partial order $\preceq$, the partial order $\preceq$ is a decidable wqo, and the set of minimal predecessors is computable. Hence, decidability follows. □

The markings of an Eos have a bounded nesting depth. For general object nets there is no bound for the nesting depth and we know that the argument as given in Theorem 6 cannot be applied since we know that even simple object nets with unbounded nesting depth are able to simulate counter programs [Köhler-Bußmeier and Heitmann, 2009]. The reasons for this lies in the fact that a partial order, defined analogously as $\preceq$, fails to be a wqo for unbounded nesting since we may have infinitely many incomparable markings. Consider e.g. the following sequence:

$$\begin{aligned} \mu_1 &= 2p_1[] \\ \mu_2 &= p_1[2p_2[]] \\ \mu_3 &= p_1[p_2[2p_3[]]] \\ &\vdots \end{aligned}$$

7.2 Reachability for Conservative EOS

In the following we show that reachability remains undecidable even if we restrict Eos to conservative typings. We can reuse the translation of a counter program statement cmd_k into a net fragment $\widehat{N}(cmd_k)$ from above (with slight modifications). The Figure 10 shows the direct translation of a counter program statement cmd_k into a net fragment $\widehat{N}(cmd_k)$. The fragments share the common counter places cnt_j for all $j \in \{1, \dots, m\}$ and the state places q_k for all $k \in \{1, \dots, n\}$. The major extension compared to Figure 6 is the global place *control*, which is a side condition to the test transition =0, and the transition reduce.

Definition 6. *Let* $CP = cmd_1; \dots; cmd_{n-1};$ halt *be a* m*-counter program. The simulating* Eos *is defined as*

$$OS(CP) = (\widehat{N}, \mathcal{N}, d, l, \mu_0) \quad with \quad \mathcal{N} = \{\bullet, N_c\}$$

- *The system net* $\widehat{N}$ *is defined as the union of all net fragments* $\widehat{N}(cmd_k)$ *for* $k \in \{1, \dots, n\}$ *as in Figure 10.*
- *The object net* N_c *is given as in Figure 10.*
- *The synchronisation* l *is given by the labels* inc, dec *and* is-pos.

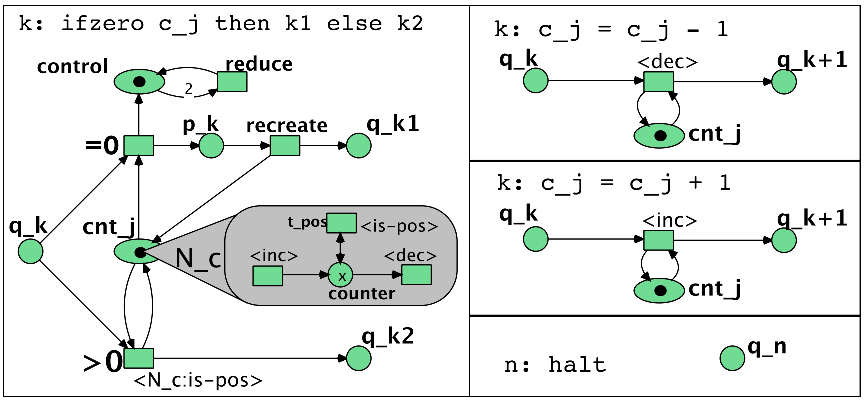


Fig. 10. The Eos-translation of counter commands

- *The type $d(p)$ of the system net places is $d(cnt_j) = d(control) = N_c$ for $j \in \{1, \dots, m\}$ and $\bullet$ otherwise.*
- *The initial marking is $\mu_0 = q_1[] + control[\mathbf{0}] + \sum_{j=1}^{m} cnt_j[\mathbf{0}]$.*

The marking μ of the Eos $OS(CP)$ encodes the configuration $C = (k, n_1, \dots, n_m)$, denoted $\mu = \mu(C)$, if the following holds:

1. The place q_k is marked with a black token.
2. For each counter c_j the counter place cnt_j is marked with one net-token that has n_j tokens on the place $counter$.
3. The place $control$ is marked with one empty net-token.

Formally: $$\mu((k, n_1, \dots, n_m)) := q_k[] + control[\mathbf{0}] + \sum_{j=1}^{m} cnt_j[n_j \cdot counter]$$

We now show how one step of the counter automaton and the test for zero can be simulated by the Eos $OS(CP)$. The simulation provided here is a very weak one, in the sense that the simulation might make wrong guesses about the test on zero, but all misguesses are "remembered" in the marking. Nevertheless for each computation $C_0 \xrightarrow[CP]{*} C$ there is one corresponding firing sequence $\mu(C_0) \xrightarrow[OS(CP)]{*} \mu(C)$. Additionally we have the property that for each firing $\mu(C_0) \xrightarrow[OS(CP)]{*} \mu$ such that μ does not correspond to any configuration then no marking reachable from μ ever will do so due to the logging on the place $control$.

Lemma 9. *Each computation of CP is simulated weakly by a firing sequence of $OS(CP)$ in the following sense:*

$$C \xrightarrow[CP]{*} C' \iff \mu(C) \xrightarrow[OS(CP)]{*} \mu(C')$$

Proof. The basic idea is to model the counter c_j by an object net on the place cnt_j (cf. Figure 10) and its value by n_j tokens on the place $counter$. Incrementation and decrementation are directly simulated via one synchronised event. The test whether the counter is greater than zero is realised by a synchronisation of the system net transition >0 with the object net transition is-pos.

For the test on zero we have to guess. First note, that different from Lemma 5 transition =0 can fire even if the object net on cnt_j is not empty. The key issue is the marking of the control places. It records the misguesses and this information is never lost. Transition =0 only guesses that the counter is equal zero and this guess might be wrong. But this guess is logged on the place *control* which is initially marked by an unmarked object net. The whole object net is transferred from cnt_j to *control* since place p_k has a different type. Transition reduce combines the old and the new net-token on *control* into one. Transition recreate generates a new object net with an empty marking (as required) on place cnt_j.

We have the property that in every reachable marking all the net-tokens on place *control* have the empty marking if and only if all guesses have been right during the simulation: The Eos starts with the marking $\mu_0 = \mu(C_0)$. When all guesses have been right during the simulation then the resulting marking perfectly reflects the configuration C. But after the first wrong guess we never reach a marking μ such that it is a configuration marking $\mu(C)$ for some C since we can never get rid of the tokens in the net-token on the place *control*. □

The simulation is called weak since in general due to non-deterministic choices for each $\mu(C)$ we have several firing sequences, i.e. the set

$$RS\big(OS(CP), \mu(C_0)\big) = \{\mu \mid \mu(C_0) \xrightarrow[OS(CP)]{*} \mu\}$$

contains also markings that are generated by incorrect simulations. But if consider the intersection of this set with the set of all configurations, i.e. the set

$$RS\big(OS(CP), \mu(C_0)\big) \cap \{\mu(C) \mid C \text{ is a configuration of } CP\},$$

then we obtain exactly those reachable marking that corresponds to correct simulations of the counter program.

Therefore this weak simulation suffices to establish the undecidability of the reachability problem.

Theorem 7. *Reachability is undecidable for conservative* Eos. *It is even undecidable for pure* Eos *with conservative typing and undecidable for minimal* Eos *with conservative typing.*

Proof. It is well-known that it is undecidable for a counter program CP whether the halting configuration with empty counters, i.e. $(n, 0, ..., 0)$ is reachable. If reachability is decidable for Eos, then deciding whether $\mu(1, 0, \ldots, 0) \xrightarrow[OS(CP)]{*} \mu(n, 0, \ldots, 0)$ holds for $OS(CP)$ decides whether $(1, 0, \ldots, 0) \xrightarrow[A]{*} (n, 0, \ldots, 0)$ terminates successfully for CP – clearly a contradiction to the undecidability of the halting problem, so reachability is undecidable for Eos. Since $OS(CP)$ is a minimal Eos reachability cannot be decidable for this class. The same argument applies if we exchange the object net $\bullet$ with another object net $N_\emptyset$ without places or transitions. Then the simulating Eos is pure. □

It is an open question whether the reachability problem is decidable for conservative, minimal and pure (i.e. unary) Eos. The interesting case for reachability of unary Eos considers unbounded object nets, since reachability is decidable for semi-bounded Eos (see Thm. 14).

Due to the weak form of simulation not everything becomes undecidable for Eos with conservative typing. In fact, as shown in Theorem 6 boundedness is decidable for conservative Eos.

7.3 The Liveness Problem for EOS

We define the liveness problem for Eos analogously to that of p/t nets: For the liveness problem one has to decide whether all events $\theta \in \Theta$ of a given Eos OS are live. An event θ is live if for all markings μ reachable from μ_0 there exists a marking μ' reachable from μ that enables θ.

Theorem 8. *Liveness is undecidable for* Eos*. More precisely, liveness is undecidable even for pure* Eos *as well as for minimal* Eos *(even with the restriction to conservative typing).*

Proof. We extend the construction of Theorem 7. We will show that if we can decide liveness for a given event $\widehat{\tau}[\vartheta]$, then we can decide reachability of the configuration $C = (n, 0, \ldots, 0)$ (i.e. proper termination with empty counters) which is an undecidable problem.

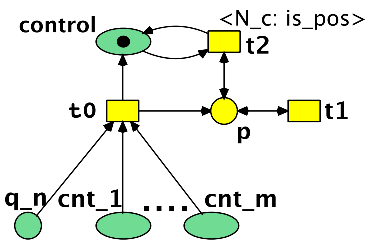


Fig. 11. Eos-fragment for Theorem 8

We use the translation of a counter program given in Figure 10 and add the fragment shown in Figure 11 consisting of the place p and the transitions t_0, t_1, and t_2. The resulting Eos is called $OS_{live}(CP)$.

We show that the configuration $(n, 0, \ldots, 0)$ is reachable in the counter program iff $t_1[\emptyset]$ is live in $OS_{live}(CP)$ and $t_2[N_c \mapsto t_{pos}]$ is not.

($\Rightarrow$) Assume that $(n, 0, \ldots, 0)$ is a reachable configuration. Then we know that in $OS(CP)$ we reach the configuration:

$$\mu((n, 0, \ldots, 0)) = q_n[] + \sum_{j=1}^{m} cnt_j[\mathbf{0}]$$

In the marking $\mu((n, 0, \ldots, 0))$ the transition t_0 is enabled and after firing t_0 the place p is marked with a token. Note that after firing of t_0 no other transition than t_1, t_2, and reduce is enabled any more.

Whenever p is marked transition t_1 is constantly enabled and therefore live. Whenever the simulation was perfect until now the global place *control* contains only net-tokens with empty marking. Firing of t_0 adds another net-token with empty marking to this place. Since transition t_2 synchronises via is-pos it is not enabled, i.e. it is dead.

Whenever at least one simulation guess was wrong then the global place *control* contains at least one net-token with a non-empty marking and transition t_2 is constantly enabled and therefore live.

($\Leftarrow$) Assume that $t_1[\emptyset]$ is live and $t_2[N_c \mapsto t_{pos}]$ is not. Since $t_1[\emptyset]$ is live we know that p is marked and therefore transition t_0 must have fired. This means that the simulation has reached the final statement since t_0 fires only if q_n is marked.

Since $t_2[N_c \mapsto t_{pos}]$ is not live, we know that the simulation must have correct and that t_0 has fired in a configuration marking, where all counters were zero.

Thus any algorithm for the Eos-liveness problem decides the halting problem for counter programs. □

7.4 Weak Simulation of Inhibitor Nets by Conservative EOS

In the following we show that reachability remains undecidable even if we restrict Eos to conservative typings. We can reuse the translation of a inhibitor net in Lemma 6. The simulation provided for conservative Eos is a very weak one, in the sense that the simulation might make wrong guesses about the test on zero, but all misguesses are stored in the marking till the end. Nevertheless for each firing sequence $m \xrightarrow{*} m'$ in the inhibitor net there is one corresponding sequence $\tilde{\mu}(m) \xrightarrow{*} \tilde{\mu}(m')$ in the simulating Eos. Additionally we have the property that for each firing $\tilde{\mu}(m) \xrightarrow{*} \mu$ such that μ does not correspond to any marking in the inhibitor net then no marking reachable from μ ever will do so.

Lemma 10. *For each inhibitor net N^* there is a conservative* Eos *$OS(N^*)$ that has the following property:*

$$m \xrightarrow[N^*]{*} m' \iff \tilde{\mu}(m) \xrightarrow[OS(N^*)]{*} \tilde{\mu}(m')$$

Proof. Let us consider a inhibitor net given as $N^* = (P^*, T^*, F^*, F_r^*, m_0)$, where $F_r^* \subseteq P^* \times T^*$ describes the inhibitor arcs.

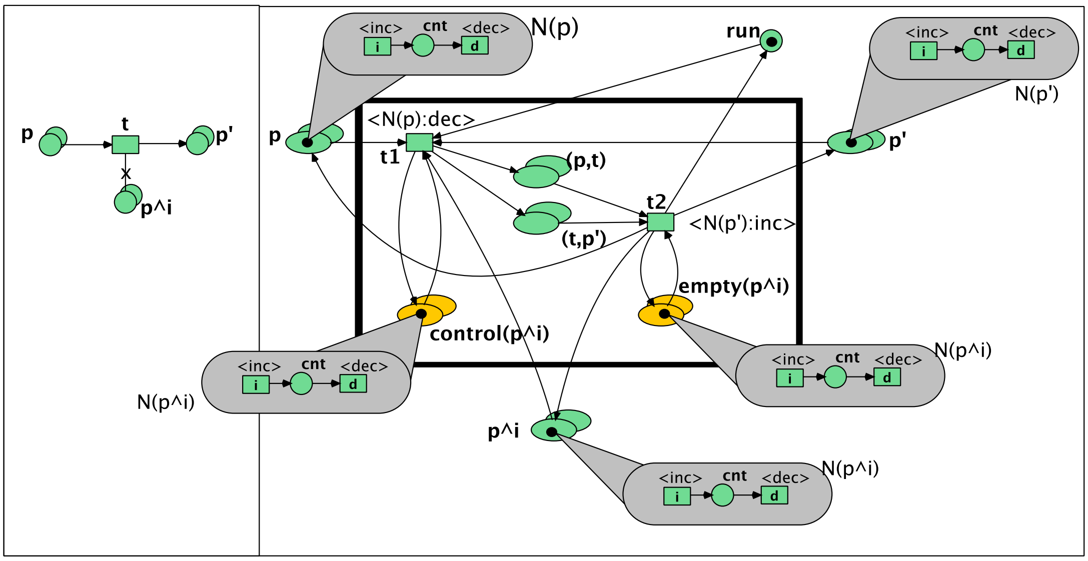


Fig. 12. The conservative EOS-translation of inhibitor nets

The simulating EOS $OS(N^*)$ (cf. Figure 12) is obtained by minor modifications from the EOS $OS_{strong}(N^*)$ from Lemma 6 (cf. Figure 7): In addition to the places in $OS_{strong}(N^*)$ we add the system net places $\mathsf{control}(p)$ and $\mathsf{empty}(p)$ with $d(\mathsf{control}(p)) = d(\mathsf{empty}(p)) = N(p)$ for each place $p \in P^*$. For each inhibitor place p^i we add $\mathsf{control}(p^i)$ as a side condition to t_1 and $\mathsf{empty}(p^i)$ to t_2. All the places (p^i, t) are removed.

$$\widehat{P} := P^* \cup F^* \cup \{\mathsf{run}\} \cup \{\mathsf{control}(p), \mathsf{empty}(p) \mid p \in P^*\}$$

The definition of $\tilde{\mu}(m)$ has to be adjusted, too: We say that a nested marking μ encodes a marking m of N^* whenever μ contains exactly one net-token on each place $p \in P^*$ (and the net-token on p has exactly $m(p)$ tokens on its place cnt) and one empty net-token on each $\mathsf{control}$ place and each empty place:

$$\begin{aligned}\tilde{\mu}(m) &:= \mu(m) + \mathsf{control}(p)[\mathbf{0}] + \mathsf{empty}(p)[\mathbf{0}] \\ &= \mathsf{run}[] + \textstyle\sum_{p \in P^*} p[m(p) \cdot \mathsf{cnt}_{d(p)}] + \mathsf{control}(p)[\mathbf{0}] + \mathsf{empty}(p)[\mathbf{0}]\end{aligned}$$

As before, the initial marking is defined as the encoding of m, i.e. $\mu_0 := \tilde{\mu}(m_0)$. By construction, the simulating EOS $OS(N^*)$ is conservative.

Assume that we have corresponding marking $\tilde{\mu}(m)$ and m enables t in the inhibitor net. As before $t_1[\vartheta_1]$ can fire if there enough tokens in the preset. Since, the run-place is emptied the simulation of other transitions is disabled then. As before $t_2[\vartheta_2]$ generates the correct successor marking. For each inhibitor place p^i the event $t_1[\vartheta_1]$ combines the net-token on p^i with that from $\mathsf{control}(p^i)$. Since t is activated in N^* the place p^i must be empty and therefore the net-token on p^i in the EOS is empty, too. After firing $t_1[\vartheta_1]$ we have an empty net-token on each $\mathsf{control}(p^i)$ again. After that, $t_2[\vartheta_2]$ regenerates an empty net-token on each

p^i with the help of the empty net-token empty(p^i): For each p^i two net-tokens are generated – one on p^i and one on empty(p^i) – and the firing rule ensures that both are empty again. Additionally, $t_2[\vartheta_2]$ puts a token back on the run-place and the simulation can continue.

However, it is not guaranteed that the inhibitor places are empty whenever $t_1[\vartheta_1]$ is enabled, i.e. for the test on zero we have to guess. What happens if some inhibitor place p^i is not empty when $t_1[\vartheta_1]$ fires? Note that $t_2[\vartheta_2]$ is still enabled but the event will produce a marking μ_1 which has a non-empty net-token on control(p^i). After that, $t_2[\vartheta_2]$ produces an empty net-token puts a on the inhibitor place p^i and puts a token back on run-place.

The important aspect is, that the resulting marking μ does not correspond to a marking of N^* since the is no m such that $\mu = \tilde{\mu}(m)$ holds since at least one control-place is marked with a non-empty token. And even more important, we can never get rid of these tokens again, since the tokens in the net-token on a control-place are never removed.

So, we have that all the net-tokens on control-places have the empty marking if and only if all guesses on the emptiness of inhibitor places have been right during the simulation: When all guesses have been right during the simulation then the resulting marking perfectly reflects the marking m. But after the first wrong guess we never reach a marking μ such that it is a configuration marking $\tilde{\mu}(m)$ for some m since we can never get rid of the tokens in the net-token on the places control(p). □

The reduction from the reachability problem to the liveness problem is also possible for conservative Eos.

Lemma 11. *The reachability problem is reducible to the liveness problem for conservative* Eos.

Proof. Figure 13 shows the conservative Eos $OS(N^*)$ with the property that if one can decide liveness for $OS(N^*)$ then one can decide reachability of the empty marking in the inhibitor net N^*. The construction is quite similar to the one in Lemma 7: To each control place control(p) we add the transitions t_1 and t_2. Whenever $t_1[\vartheta_1]$ is enabled then the net-token is not empty. Note, that the sequence $t_1[\vartheta_1]t_2[\vartheta_2]$ does not change the marking.

As before liveness of $t_0[\vartheta_0]$ indicates reachability of the empty marking $\mathbf{0}$ in the inhibitor net N^* under the the assumption that all guesses habe been made correct. So, we have to express the condition that all guesses about imhibitor test have been right during the weak simulation in terms of liveness: If all guesses about imhibitor tests on place p have been right during the weak simulation, then $t_1[\vartheta_1]$ is dead for all ϑ_1. Conversely, if one guess has been wrong, then $t_1[\vartheta_1]$ is live for some ϑ_1. Therefore, the simulation is correct iff $t_1(p)[\vartheta_1]$ is dead for all $p \in P^*$.

Therefore, the empty marking $\mathbf{0}$ is reachable in the inhibitor net N^* iff for all $p \in P^*$ we have that $t_1[\vartheta_1]$ is not live for all ϑ_1 and $t_0[\vartheta_0]$ is live for some ϑ_0. □

Therefore, liveness is undecidable even for conservative Eos.

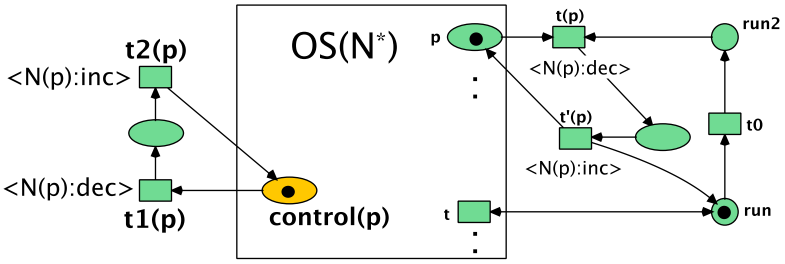


Fig. 13. Conservative EOS-Reduction from the **0**-Reachability to Liveness of t_0

Theorem 9. *Reachability and liveness are undecidable for conservative* EOS.

8 Decidability Results for Generalised State Machines

For each EOS there is an obvious construction of a p/t net, called the reference net, which is constructed by taking as the set of places the disjoint union of all places and as the set of transitions the synchronisations. Since the places of all nets in $\mathcal{N}$ are disjoint by definition, the projections $(\Pi^1(\mu), (\Pi^2_N(\mu))_{N\in\mathcal{N}})$ can be identified with the multiset:

$$\mathrm{RN}(\mu) := \Pi^1(\mu) + \sum_{N\in\mathcal{N}} \Pi^2_N(\mu)$$

Note that $\mathrm{RN}(\mu)$ is another representant of the equivalence class $[\mu]_\cong$.

Definition 7. *Let* $OS = (\widehat{N}, \mathcal{N}, d, l, \mu_0)$ *be an* EOS. *The* reference net $\mathrm{RN}(OS)$ *is defined as the p/t net:*

$$\mathrm{RN}(OS) = \left(\left(\widehat{P} \cup \bigcup\nolimits_{N\in\mathcal{N}} P_N\right), \Theta, \mathbf{pre}^{\mathrm{RN}}, \mathbf{post}^{\mathrm{RN}}, \mathrm{RN}(\mu_0)\right)$$

where $\mathbf{pre}^{\mathrm{RN}}$ *(and analogously:* $\mathbf{post}^{\mathrm{RN}}$*) is defined by:*

$$\mathbf{pre}^{\mathrm{RN}}(\widehat{\tau}[\vartheta]) = \mathbf{pre}(\widehat{\tau}) + \sum\nolimits_{N\in\mathcal{N}} \mathbf{pre}_N(\vartheta(N))$$

The net is called *reference net* because it behaves as if each object net would have been accessed via pointers and not like a value: A black token on a system net place $\widehat{p}$ is interpreted as a pointer to the object $\widehat{N} = d(\widehat{p})$ where each object net has exactly one instance but several pointers referring to it.

Theorem 10. *Let OS be an* EOS. *Every event* $\widehat{\tau}[\vartheta]$ *that is activated in OS for* (λ, ρ) *is so in* $\mathrm{RN}(OS)$*:*

$$\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu' \quad\Longrightarrow\quad \mathrm{RN}(\mu) \xrightarrow[\mathrm{RN}(OS)]{\widehat{\tau}[\vartheta]} \mathrm{RN}(\mu')$$

Proof. Whenever $\widehat{\tau}[\vartheta]$ is activated in μ the enabling condition ϕ holds. This implies that $\Pi^1(\mu)$ enables $\widehat{\tau}$ and $\Pi^2_N(\mu)$ enables $\vartheta(N)$ for each $N \in \mathcal{N}$. Since all the places are disjoint $\mathrm{RN}(\mu)$ is isomorphic to the projections $\Pi(\mu)$ and this implies that the multiset sum $\widehat{\tau} + \sum_{N\in\mathcal{N}} \vartheta(N)$ is enabled which is equivalent to the enabling in $\mathrm{RN}(OS)$. Analogously one can observe that the effect on $\Pi^1(\mu)$ an on the $\Pi^2_N(\mu)$ is the same which implies that the successor marking in $\mathrm{RN}(OS)$ is $\mathrm{RN}(\mu')$. □

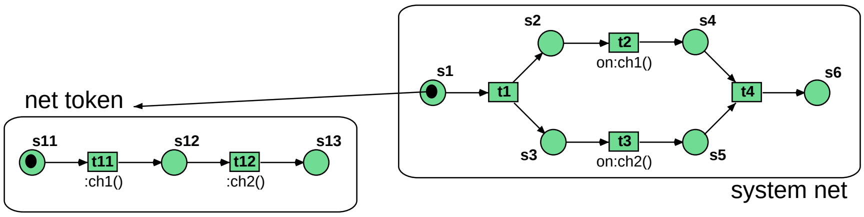


Fig. 14. The α-centauri EOS

The converse is not true in general, which can be demonstrated using the EOS in Fig. 14 known as the α-centauri example, cf. [Valk, 1998]. Initially we have $\mu_0 = \widehat{s}_1[s_{11}]$. In the reference net we have the initial marking $\mathrm{RN}(\mu_0) = \widehat{s}_1 + s_{11}$ which activates the firing sequence:

$$(\widehat{s}_1 + s_{11}) \xrightarrow{\widehat{t}_1[\vartheta_{id}]} (\widehat{s}_2 + \widehat{s}_3 + s_{11}) \xrightarrow{\widehat{t}_2[t_{11}]} (\widehat{s}_4 + \widehat{s}_3 + s_{12}) \xrightarrow{\widehat{t}_3[t_{12}]} (\widehat{s}_4 + \widehat{s}_5 + s_{13})$$

It is easy to see that in the EOS we can fire only a prefix, depending on the choice of the modes. The first mode assigns the token on s_{11} to the net-token on $\widehat{s}_3$:

$$\widehat{s}_1[s_{11}] \xrightarrow{\widehat{t}_1[\vartheta_{id}]} \widehat{s}_2[\mathbf{0}] + \widehat{s}_3[s_{11}]$$

The second mode assigns the token on s_{11} to the net-token on $\widehat{s}_2$:

$$\widehat{s}_1[s_{11}] \xrightarrow{\widehat{t}_1[\vartheta_{id}]} \widehat{s}_2[s_{11}] + \widehat{s}_3[\mathbf{0}] \xrightarrow{\widehat{t}_2[t_{11}]} \widehat{s}_4[s_{12}] + \widehat{s}_3[\mathbf{0}]$$

Since the effect in the object net is only local, $\widehat{t}_3[t_{12}]$ is not activated. So $w = \widehat{t}_1[\vartheta_{id}] \cdot \widehat{t}_2[t_{11}] \cdot \widehat{t}_3[t_{12}]$ is a possible firing sequence for the reference net, but not for the object net system.

From Theorem 10 and the above the following property follows.

Corollary 1. *Let OS be an* EOS. *If μ is reachable from μ_0, then* $\mathrm{RN}(\mu)$ *is reachable from* $\mathrm{RN}(\mu_0)$. *The reverse does not hold in general.*

So, we obtain only a sufficient condition for non-reachability: The marking μ is not reachable from μ_0 whenever $\mathrm{RN}(\mu)$ is not reachable from $\mathrm{RN}(\mu_0)$.

Fortunately, many practical models are *Generalised State Machines* and this sufficient condition can be strengthened to a necessary one for these.[7] An Eos *OS* is a *generalised state machine* (GSM) iff for all $\widehat{t}$ there is either exactly one place in the preset and one in the postset typed with the object net N or there are no such places:

$$\forall N \in \mathcal{N} : \forall \widehat{t} \in \widehat{T} : \left|\{\widehat{p} \in {}^{\bullet}\widehat{t} \mid d(\widehat{p}) = N\}\right| = \left|\{\widehat{p} \in \widehat{t}^{\bullet} \mid d(\widehat{p}) = N\}\right| \leq 1 \tag{11}$$

and the initial marking μ_0 has at most one net-token of each type, denoted as $\psi_{gsm}(\mu_0)$, where:

$$\psi_{gsm}(\mu) := \forall N \in \mathcal{N} : \sum_{\widehat{p} \in \widehat{P}, d(\widehat{p}) = N} \Pi^1(\mu)(\widehat{p}) \leq 1 \tag{12}$$

Obviously every p/t-like Eos is a generalised state machine since $d(\widehat{p}) = \bullet$ for all $\widehat{p}$. In addition, generalised state machines are conservative Eos.

For generalised state machines we can strengthen Theorem 10.

Theorem 11. *Let OS be an* Eos *with the generalised state machine property. A transition $\widehat{\tau}[\vartheta]$ is activated in OS for (λ, ρ) iff it is in* Rn(*OS*)*:*

$$\mu \xrightarrow[OS]{\widehat{\tau}[\vartheta](\lambda,\rho)} \mu' \quad \iff \quad \text{Rn}(\mu) \xrightarrow[\text{Rn}(OS)]{\widehat{\tau}[\vartheta]} \text{Rn}(\mu')$$

Proof. By (12) the property holds initially, i.e. $\psi_{gsm}(\mu_0)$ is true. It is easy to observe that that the property $\psi_{gsm}(\mu)$ remains true in all reachable markings, since whenever there is at most one net-token for each object net in marking μ, then (11) implies that there are equally many net-tokens in the successor marking μ'.

Therefore in each reachable marking μ we have for each object net N that is present in the initial marking exactly one marked system net place $\widehat{p}_N$ which contains the net-token of type N.

In this case all tokens in the projection $\Pi^2_N(\mu)$ belong to the marking of the net-token on $\widehat{p}_N$. The net-token can be reconstructed as $\widehat{p}_N[\Pi^2_N(\mu)]$.

Therefore, we can uniquely reconstruct μ from Rn(μ) and reachability in the net Rn(*OS*) is a necessary and sufficient condition for reachability. □

A generalised state machine *OS* is therefore isomorphic with its reference net Rn(*OS*).

Theorem 12. *The reachability problem is decidable for p/t-like* Eos *and for generalised state machines.*

[7] This class was introduced in [Köhler and Rölke, 2005] to study the relationship between different variants of nets-within-nets semantics. Here we study it of Eos in the context of decidability questions.

Proof. In Theorem 11 it is shown that if the system net is a generalised state machine then reachability in the net $\text{Rn}(OS)$ is a necessary and sufficient condition for reachability. Since reachability is decidable for p/t nets, it is so for generalised state machines.

Decidability for p/t-like Eos holds since each p/t-like Eos is a generalised state machine. □

Theorem 13. *The liveness problem is decidable for p/t-like* Eos *and for generalised state machines.*

Proof. For p/t nets it is well known that liveness is reducible to reachability (cf. Peterson, 1981, chapter 5). Applying Theorem 11 the same reduction from liveness to reachability can be done for $\text{Rn}(OS)$ to decide liveness of a given event $\widehat{\tau}[\vartheta]$. The result follows since reachability is decidable for p/t nets. □

From a modelling point of view these results are interesting since in many scenarios net-tokens model physical entities which are neither cloned, combined, created nor destroyed. These models therefore have the generalised state machine property. From a more theoretical point of view the correspondence of each generalised state machine OS with its reference net $\text{Rn}(OS)$ allows to simplify notations considerably – at the price of limiting the expressiveness. For these reasons some formalism, like e.g. [Valk, 1998], [Bednarczyk et al., 2004], or [Lomazova et al., 2006], are initially restricted to generalised state machines. For our analysis we have chosen to study the general case to obtain more insights in the models properties and their expressiveness.

9 Boundedness and Safe EOS

In the following we study how boundedness restrictions on either the system net or the object nets influence decidability results. We show that reachability is decidable whenever the object nets are bounded.

9.1 Semi-Bounded EOS

For p/t nets it is well known there are only finitely many reachable markings iff the net is n-safe for some $n \in \mathbb{N}$. A p/t net is called n-safe with $n \in \mathbb{N}$ if in every reachable marking there at most n tokens on each place: $\forall m \in RS(m_0) : \forall p \in P : m(p) \leq n$. A net that is n-safe for some n is also called bounded. The following property is well-known for p/t nets.

Lemma 12. *The set of reachable markings of a p/t net N is finite iff N is n-safe for some n.*

Proof. If the set of reachable markings is finite then N is n-safe for $n := \max\{m(p) \mid p \in P, m \in RS(m_0)\}$. If N is n-safe then $|RS(m_0)| \leq (n+1)^{|P|}$. □

An EOS is N-*bounded* if $\Pi^2_N(\mu)$ is bounded in each reachable marking μ. An EOS is *semi-bounded* if it is bounded for all object nets $N \in \mathcal{N}$.

Note that it is possible that an EOS is N-bounded but N (considered as a p/t net in isolation) is not bounded for the initial marking $\Pi^2_N(\mu_0)$. Any unbounded object net, where each transition is synchronised with a dead system net transition is an example.

Theorem 14. *The reachability problem is decidable for a semi-bounded* EOS.

Proof. The set of $\Pi^2_N(\mu)$ for all reachable markings μ is bounded by definition. Let B_N be its least upper bound. Note that since the object nets are bounded, so are the firing modes λ and ρ. Define the (finite) set of places $P_{os} = \{(\widehat{p}, M) \mid \widehat{p} \in \widehat{P} \wedge M \leq B_{d(\widehat{p})}\}$ and the (finite) set of transitions

$$T_{os} = \{(\theta, \lambda, \rho) \mid \phi(\theta, \lambda, \rho) \wedge \forall N \in \mathcal{N} : \Pi^2_N(\lambda) \leq B_N \wedge \Pi^2_N(\rho) \leq B_N\}.$$

Define $\mathbf{pre}((\theta, \lambda, \rho))((\widehat{p}, M)) = \lambda(\widehat{p}, M)$ and $\mathbf{post}((\theta, \lambda, \rho))((\widehat{p}, M)) = \rho(\widehat{p}, M)$. The initial marking is $M_0((\widehat{p}, M)) = \mu_0(\widehat{p}, M)$. It is easy to observe that the p/t net $(P_{os}, T_{os}, \mathbf{pre}, \mathbf{post}, M_0)$ simulates the semi-bounded EOS. □

Note that the symmetric boundedness condition, i.e. a bounded system net, does not lead to any real restriction since the EOS $OS_{strong}(CP)$, which simulates counter programs, has a bounded system net.

9.2 Boundedness for Safe EOS

In the following we study EOS, where the system net and all the object nets are bounded. Then we know from Thm. 14 that reachability is decidable. So we interested in the complexity of the decision procedure.

A p/t net is called *safe* if it is 1-safe. Therefore, each reachable marking of a safe net is a set and we have $|RS(m_0)| \leq 2^{|P|}$, i.e. the number of subsets of P.

We extend the definition of 1-safe nets from p/t nets to EOS. We can identify at least the following variants for the definition of safeness:

Definition 8 (Safe EOS). *Let OS be an* EOS.

- *OS is safe(1) iff all reachable markings are sets.*

$$\forall \mu \in RS(OS) : \forall \widehat{p}[M] : \mu(\widehat{p}[M]) \leq 1$$

- *OS is safe(2) iff for all reachable markings there is most one token on each system net place:*

$$\forall \mu \in RS(OS) : \forall \widehat{p} \in \widehat{P} : \Pi^1(\mu)(\widehat{p}) \leq 1$$

– *OS is safe(3) iff for all reachable markings there is most one token on each system net place and each net-token is safe:*

$$\forall \mu \in RS(OS) : \forall \widehat{p} \in \widehat{P} : \Pi^1(\mu)(\widehat{p}) \leq 1 \wedge \\ \forall N \in \mathcal{N} : \forall p \in P_N : \forall \widehat{p}[M] \sqsubseteq \mu : M(p) \leq 1$$

– *OS is safe(4) iff for all reachable markings there is most one token on each place (w.r.t. projections):*

$$\forall \mu \in RS(OS) : \forall \widehat{p} \in \widehat{P} : \Pi^1(\mu)(\widehat{p}) \leq 1 \wedge \\ \forall N \in \mathcal{N} : \forall p \in P_N : \Pi^2_N(\mu)(p) \leq 1$$

An OS is called *system-safe* if it is safe(2), it is called *safe* if it is safe(3), and it is called *strongly safe* if it is safe(4). The reason for this will be justified by the following discussion.

Lemma 13. *If an* Eos *is safe(4) then it is safe(3).*
If an Eos *is safe(3) then it is safe(2). If an* Eos *is safe(2) then it is safe(1).*

Proof. If an Eos is safe(4) then we have to show that $\Pi^2_N(\mu)(p) \leq 1$ implies $\forall \widehat{p}[M] \sqsubseteq \mu : M(p) \leq 1$. This follows from the fact that $\Pi^2_N(\mu) \leq M$ holds for all $\widehat{p}[M] \sqsubseteq \mu$. Therefore OS is safe(3).

If an Eos is safe(3) then it is safe(2) since the first definition directly implies the latter.

If an Eos is safe(2) then we know from $\Pi^1(\mu)(\widehat{p}) \leq 1$ that whenever there is an addend $\widehat{p}[M]$ in μ then there is at most one, i.e. $\forall \widehat{p}[M] : \mu(p[M]) \leq 1$ and OS is safe(1). □

The converse implications of Lemma 13 are not true. Examples are given in Figure 15, 16 and 17.

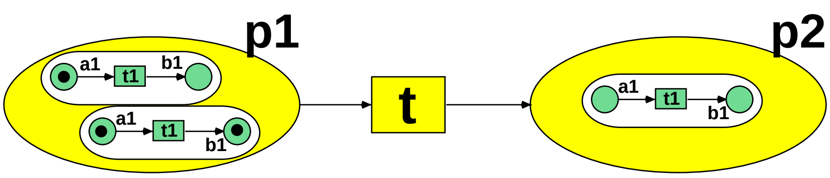


Fig. 15. Eos which is safe(1), but not safe(2)

The motivation for the variant safe(1) is clear: For a safe p/t net all reachable markings are sets and by definition all reachable markings of a safe(1) Eos are sets, too. Using Lemma 3 this holds for safe(2), safe(3), and safe(4).

Lemma 14. *If an* Eos *is safe(2), safe(3), or safe(4) then its first projection to the system net level $\Pi^1(OS)$ is a safe p/t net.*

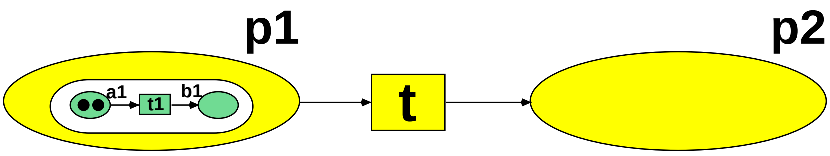


Fig. 16. EOS which is safe(2), but not safe(3)

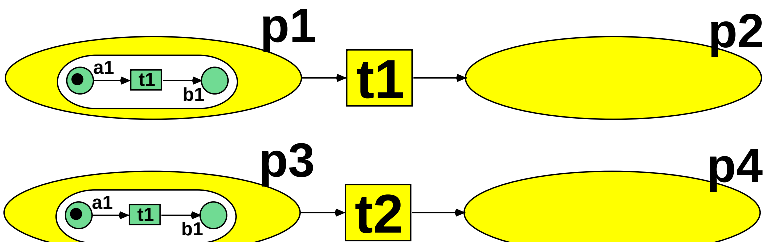


Fig. 17. EOS which is safe(3), but not safe(4)

Proof. By Lemma 3 we have:

$$\mu \xrightarrow[OS]{\widehat{t}[\vartheta]} \mu' \Longrightarrow \Pi^1(\mu) \xrightarrow[\Pi^1(OS)]{\widehat{t}} \Pi^1(\mu')$$

If an EOS is safe(2) then we have $\Pi^1(\mu)(p) \leq 1$ for all reachable μ. Thus the system net $\Pi^1(OS)$ is safe.

If an EOS is safe(4) then it is safe(3) by Lemma 13 and thus its first projection is a safe p/t net. Analogously, if an EOS is safe(3) then it is safe(2). □

Note that Lemma 14 does not hold for safeness(1) as can be seen by the example in Figure 15.

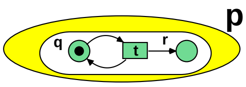
In contrast to p/t nets not all notions of safeness are equivalent to finiteness of the state space.

Theorem 15. *If an* EOS *is safe(3) or safe(4) then its set of reachable markings is finite.*

Proof. We known already by Lemma 13 that if an EOS is safe(4) then it is safe(3).

Assume that an given EOS is safe(3). Define $k := |\widehat{P}|$ and l as the maximum of $|P_N|$ for all $N \in \mathcal{N}$. Then by definition of safe(3) we have at most 2^l different net-tokens for each object net. Additionally – by definition of safe(3) – each system net place is either unmarked or marked with one of these net-tokens, i.e. we have $(1 + 2^l)^k \in O(2^{kl})$ different markings. □

The property of Theorem 15 does no longer hold for safe(1) and safe(2) EOS: The following EOS is safe(2) – and therefore also safe(1) – and has the following set of reachable markings which is infinite:



$$RS(OS) = \{\widehat{p}[q + k \cdot r] \mid k \in \mathbb{N}\}$$

If we like to preserve as many aspects of safeness the negative results concerning Lemma 14 and Thm. 15 argue in favour of the two variants safe(3) and safe(4).

For generalised state machines we can strengthen Lemma 13:

Theorem 16. *A generalised state machine is safe(3) iff it is safe(4).*

Proof. We known already by Lemma 13 that if an Eos is safe(4) then it is safe(3).

Assume that the Eos is safe(3). Then we have $M(p) \leq 1$ for all $\widehat{p}[M] \sqsubseteq \mu$. But since OS is a GSM we know that for each $N \in \mathcal{N}$ there is at most one place $\widehat{p}$ with type $d(\widehat{p}) = N$ marked with a net-token – this property holds for the initial marking and is preserved by each firing transition of a GSM.

Therefore we have $\Pi^2_N(\mu) = M$ which implies that $M(p) = \Pi^2_N(\mu)(p) \leq 1$, i.e. the Eos is safe(4). □

For p/t nets the different notions of safeness coincide.

Theorem 17. *For p/t-like* Eos *safeness(1), safeness(2), safeness(3), and safeness(4) are equivalent.*

Proof. Since p/t-like Eos are GSM we know by Thm. 16 that safeness(3) is equivalent to safeness(4) for p/t-like Eos.

We known already by Lemma 13 that if an Eos is safe(3) then it is safe(2). Assume that the Eos is safe(2). Then we have to show that $M(p) \leq 1$ holds for all $\widehat{p}[M] \sqsubseteq \mu$: Since OS is p/t-like it has only places for black tokens. Therefore, $\mu(\widehat{p}[M]) = 1$ implies $M = \mathbf{0}$ which implies $M(p) = 0 \leq 1$, i.e.the Eos is safe(3). Thus safeness(2) is equivalent to safeness(3) for p/t-like Eos.

We known already by Lemma 13 that if an Eos is safe(2) then it is safe(1). Assume that the Eos is safe(1). Then we have $\mu(\widehat{p}[M]) \leq 1$ for all $\widehat{p}[M]$. But since OS is p/t-like it is also a GSM and we know that for each $N \in \mathcal{N}$ there is at most one place $\widehat{p}$ with type $d(\widehat{p}) = N$ marked with a net-token. Thus we have $\forall \widehat{p} \in \widehat{P} : \Pi^1(\mu)(\widehat{p}) \leq 1$, i.e. the Eos is safe(3). Thus safeness(2) is equivalent to safeness(1) for p/t-like Eos. □

Note that Theorem 17 is in fact the justification for the need of different notions for safeness given in Definition 8.

To summarise, we have introduced the different variants of safeness to apply model checking techniques. By Theorem 15 we know that in general only safe(3) and safe(4) Eos have finite of state spaces, i.e. the safe and strongly safe Eos. In fact Theorem 15 justifies our terminology.

9.3 Decidability and Complexity Issues for Safe EOS

We have seen that in general safe(1) and safe(2) Eos have infinite state spaces, since the net-tokens' markings are unbounded. Therefore it is a natural question whether boundedness is decidable for these classes.

Inhibitor Net Simulation We have seen that in general safe(1) and safe(2) Eos have infinite state spaces, since the net-tokens' markings are unbounded. Therefore it is a natural question whether boundedness is decidable for these classes. If we look at the construction in Lemma 7 and in Lemma 10 we can observe that the constructed Eos are already safe(2).

Corollary 2. *Reachability, liveness, boundedness, and coverability are undecidable for safe(1) or safe(2)* Eos.

Reachability and liveness, are undecidable for safe(1) or safe(2) conservative Eos.

Counter Program Simulation Since net-tokens are unbounded for safe(1) or safe(2) Eos one may conject that is might be possible to simulate counters.

Theorem 18. *Reachability and boundedness are undecidable for safe(1) or safe(2)* Eos.

Proof. We have shown in Lemma 5 that for each counter program CP there is the Eos $OS_{strong}(CP)$ that simulates CP. If we look at the construction of $OS_{strong}(CP)$ we observe that it is a safe(2) Eos. Since reachability is undecidable for counter programs it is for safe(2) Eos and thus also for safe(1) Eos. Analogously for boundedness. □

In accordance to this, observe that the simulation construction for inhibitor nets given in Lemma 6 is also a safe(2) Eos.

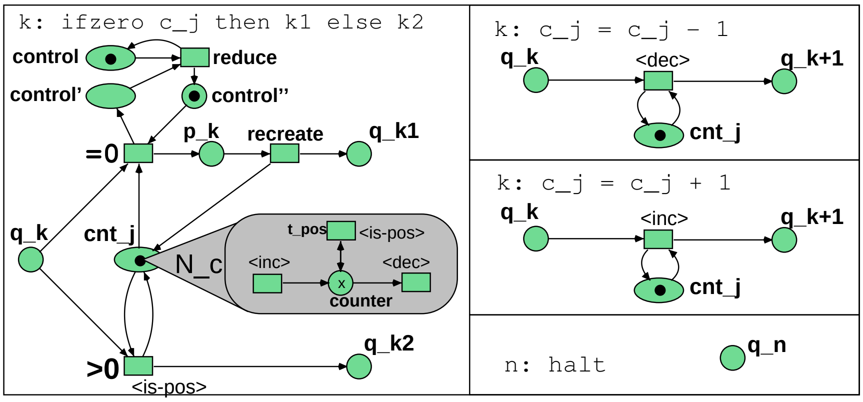


Fig. 18. The conservative Eos-translation of counter commands

We can even extend the result to the conservative case.

Theorem 19. *Reachability is undecidable for safe(1) or safe(2) conservative* Eos.

Proof. We can modify the EOS-translation of counter programs given in Figure 10 such that the EOS is safe(2). This EOS is given in Figure 18. The only non-safe place in Figure 10 is the place *control*. In the modified version there is an additional place *control*$'$ for the net-token generated by the zero test. Initially the place *control*$''$ is marked with one black token. The marking $\mu(C)$ encodes the configuration $C = (k, n_1, \ldots, n_m)$:

$$\mu((k, n_1, \ldots, n_m)) := q_k[] + \mathit{control}[\mathbf{0}] + \sum_{j=1}^{m} \mathit{cnt}_j[n_j \cdot \mathit{counter}] + \mathit{control}''[\mathbf{0}]$$

The place *control*$''$ enforces that a reduction occurs before the next test. So, there is always exactly one net-token on *control* and at most one on *control*$'$. It is easy to see that this modified version provides the same weak simulation of counter programs as the EOS $OS(CP)$ of Lemma 9. □

We have seen that liveness is undecidable for general EOS. The problem remains undecidable even for safe(1) or safe(2) EOS.

Theorem 20. *Liveness is undecidable for safe(1) or safe(2)* EOS*, even for conservative typing.*

Proof. We can re-use the construction used in Theorem 8. We start with the translation of counter commands into a safe(2) EOS given in Figure 18 and extend the fragment of Figure 11. Again we have, that if we can decide liveness for a given event $\widehat{\tau}[\vartheta]$, then we can decide reachability. □

Complexity of Safe EOS By Theorem 15 we know that safe EOS have finite state spaces, but compared to the state spaces of p/t nets they may become quite large: For p/t nets it is known that whenever there are n places, then the number of reachable states is bound by $O(2^n)$. In the proof of Thm. 15 we have seen that whenever the number of places in the system net and in all object nets is bound by n, then the number of reachable states is in $O(2^{n^2})$ – a quite drastic increase: If the system net and the object nets are quite small, say with at most $n = 10$ places, each of them has a state spaces of size 2^{10}, so we can consider them small enough to directly represented and analysed. However, their composition within the EOS generates a state space of size $O(2^{100})$. This combinatorial explosion makes it in general very hard to represent the state space explicitly.

We know that problems for safe(3) or safe(4) EOS are at least complex as the corresponding problem for p/t nets. It is a known fact that most interesting questions about the behaviour of classical 1-safe p/t nets like liveness, deadlock-freedom, and reachability are PSPACE-hard (see Esparza, 1998). This follows from the fact, first observed in [Jones et al., 1977], that a 1-safe p/t net of size $O(n^2)$ can simulate a linear bounded automaton starting on an empty tape of size n. Since the net can furthermore be constructed in polynomial time, hardness results concerning linear bounded automata carry over to 1-safe p/t nets. From there they directly carry over to safe Eos, since 1-safe p/t nets can be seen as a special kind of safe(4) Eos. Thus it is PSPACE-hard to decide reachability and liveness for safe Eos.

Theorem 21. *For safe(3) and safe(4)* Eos *the reachability and the liveness problem are* PSPACE*-hard.*

The more interesting question is therefore, if polynomial space suffices. The second part of this survey is going to characterise this kind of blow-up in complexity theoretical terms. It turns out that polynomial space is also sufficient to decide reachability and liveness (cf. Köhler-Bußmeier and Heitmann, 2010b,a for details), so both are PSPACE-complete problems.

10 Conclusion

This paper studies the Petri net formalism of elementary object net systems (Eos). Object nets are Petri nets which have Petri nets as tokens. Eos are called elementary since the nesting is restricted to two levels only. The general formalism of objects nets allows arbitrarily nested nets. The algebraic extension of objects nets, called HORNETS, even allows operations on the net-tokens, like sequential or parallel composition.

Interestingly enough, even for the restricted class of elementary object nets reachability, liveness, and boundedness are undecidable problems. (Table 1 summarises the most relevant decidability results of this paper. Here u denotes undecidability and d decidability of the problem.) Even for the class of conservative Eos – where boundedness remains decidable – the reachability and the liveness problem remain undecidable.

Table 1. Overview of Decidability Results

	Eos	conservative Eos	GSM	safe(1/2) and conservative Eos	semi-bounded Eos	safe(3/4) Eos
reachability	u	u	d	u	d	PSPACE-complete
liveness	u	u	d	u	d	PSPACE-complete
boundedness	u	d	d	d	d	always bounded

Additionally, we studied Eos that are in some sense safe systems. The discussion of safeness shows that for Eos we have at least four different variants of safeness which all coincide for p/t-like Eos. Only the class of safe(3) or safe(4) Eos have finite state spaces. The class of of safe(1) or safe(2) Eos is not really simpler as the general case as reachability and the liveness are still undecidable for them. On the other hand the LTL/CTL model checking problem for safe(3) or safe(4) Eos is as complex as the corresponding problem for p/t nets which implies that reachability and liveness are PSPACE-hard problems. (In fact they are PSPACE-complete problems as shown in Köhler-Bußmeier and Heitmann, 2010b,a.)

Bibliography

Parosh A. Abdulla, Karlis Cerans, Bengt Jonsson, and Yih K. Tsay. General decidability theorems for infinite-state systems. In *11th Annual IEEE Symposium on Logic in Computer Science, LICS*, pages 313–321, 1996.

Toshiro Araki and Tadao Kasami. Some decision problems related to the reachability problem for Petri nets. *Theoretical Computer Science*, 3:85–104, 1977.

Marek A. Bednarczyk, Luca Bernardinello, Wiesław Pawlowski, and Lucia Pomello. Modelling mobility with Petri hypernets. In Josè Luiz Fiadeiro, Peter D. Mosses, and Fernando Orejas, editors, *Recent Trends in Algebraic Development Techniques (WADT 2004)*, volume 3423 of *Lecture Notes in Computer Science*, pages 28–44. Springer-Verlag, 2004.

Nadia Busi. Mobile nets. In P. Ciancarini, A. Fantechi, and R. Gorrieri, editors, *Formal Methods for Open Object-Based Distributed Systems*, volume 139, pages 51–66. Kluwer, 1999.

Luca Cardelli, Andrew D. Gordon, and Giorgio Ghelli. Mobility types for mobile ambients. In *Proceedings of the Conference on Automata, Languages, and Programming (ICALP'99)*, volume 1644 of *Lecture Notes in Computer Science*, pages 230–239. Springer-Verlag, 1999.

Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In K. Larsen, editor, *Automata, Languages, and Programming (ICALP'98)*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer-Verlag, 1998.

Javier Esparza. Decidability and complexity of petri net problems – an introduction. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer-Verlag, 1998.

Berndt Farwer. A linear logic view of object Petri nets. *Fundamenta Informaticae*, 37(3):225–246, 1999.

Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

Serge Haddad and Denis Poitrenaud. Theoretical aspects of recursive Petri nets. In S. Donatelli and J. Kleijn, editors, *Proceedings of the 20th International Conference on Application and Theory of Petri Nets*, volume 1639 of *Lecture Notes in Computer Science*, pages 228–247. Springer-Verlag, 1999.

Kunihiko Hiraishi. PN^2: An elementary model for design and analysis of multi-agent systems. In Farhad Arbab and Carolyn L. Talcott, editors, *Coordination Models and Languages, COORDINATION 2002*, volume 2315 of *Lecture Notes in Computer Science*, pages 220–235. Springer-Verlag, 2002.

Kathrin Hoffmann, Hartmut Ehrig, and Till Mossakowski. 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*, pages 268 – 288. Springer-Verlag, 2005.

Matthias Jantzen and Rüdiger Valk. Formal properties of place/transition nets. In W. Brauer, editor, *Net Theory and Applications*, volume 84 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

N. D. Jones, L. H. Landweber, and Y. E. Lien. Complexity of some problems in petri nets. *Theoretical Computer Science*, 4:277–299, 1977.

Richard M. Karp and Raymond E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, May 1969.

Michael Köhler. *Objektnetze: Definition und Eigenschaften*. Logos Verlag, Berlin, 2004.

Michael Köhler. Reachable markings of object Petri nets. *Fundamenta Informaticae*, 79(3-4):401 – 413, 2007.

Michael Köhler and Berndt Farwer. Modelling global and local name spaces for mobile agents using object nets. *Fundamenta Informaticae*, 72(1-3):109–122, 2006.

Michael Köhler and Heiko Rölke. Concurrency for mobile object-net systems. *Fundamenta Informaticae*, 54(2-3), 2003.

Michael Köhler and Heiko Rölke. Properties of Object Petri Nets. In J. Cortadella and W. Reisig, editors, *International Conference on Application and Theory of Petri Nets 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 2004.

Michael Köhler and Heiko Rölke. Reference and value semantics are equivalent for ordinary Object Petri Nets. In P. Darondeau and G. Ciardo, editors, *International Conference on Application and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 309–328. Springer-Verlag, 2005.

Michael Köhler, Daniel Moldt, and Heiko Rölke. Modeling the behaviour of Petri net agents. In J. M. Colom and M. Koutny, editors, *International Conference on Application and Theory of Petri Nets*, volume 2075 of *Lecture Notes in Computer Science*, pages 224–241. Springer-Verlag, 2001.

Michael Köhler, Daniel Moldt, and Heiko Rölke. Modelling mobility and mobile agents using nets within nets. In W. v. d. Aalst and E. Best, editors, *International Conference on Application and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 121–140. Springer-Verlag, 2003.

Michael Köhler-Bußmeier. Hornets: Nets within nets combined with net algebra. In Karsten Wolf and Giuliana Franceschinis, editors, *International Conference on Application and Theory of Petri Nets (ICATPN'2009)*, volume 5606 of *Lecture Notes in Computer Science*, pages 243–262. Springer-Verlag, 2009.

Michael Köhler-Bußmeier and Frank Heitmann. On the expressiveness of communication channels for object nets. *Fundamenta Informaticae*, 93(1-3):205–219, 2009.

Michael Köhler-Bußmeier and Frank Heitmann. Liveness of safe object nets. In Louchka Popova-Zeugmann et al., editor, *Proceedings of the InternationaA*

Survey ofl Workshop on Concurrency, Specification, and Programming (CS&P 2010), 2010a.

Michael Köhler-Bußmeier and Frank Heitmann. Safeness for object nets. *Fundamenta Informaticae*, 101(1-2):29–43, 2010b.

Olaf Kummer. Undecidability in object-oriented Petri nets. *Petri Net Newsletter*, 59:18–23, 2000.

Olaf Kummer. *Referenznetze*. Logos Verlag, 2002.

Olaf Kummer, Frank Wienberg, Michael Duvigneau, Jörn Schumacher, Michael Köhler, Daniel Moldt, Heiko Rölke, and Rüdiger Valk. An extensible editor and simulation engine for Petri nets: Renew. In J. Cortadella and W. Reisig, editors, *International Conference on Application and Theory of Petri Nets 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 484 – 493. Springer-Verlag, 2004.

Olaf Kummer, Roxana Dietze, and Manfred Kudlek. Decidability problems of a basic class of object nets. *Fundamenta Informaticae*, 79(3-4):295–302, 2008.

Charles Lakos. A Petri net view of mobility. In *Formal Techniques for Networked and Distributed Systems (FORTE 2005)*, volume 3731 of *Lecture Notes in Computer Science*, pages 174–188. Springer-Verlag, 2005.

Jena-Luc Lambert. A structure to decide the reachability in Petri nets. *Theoretical Computer Science*, 99:79–104, 1992.

Kurt Lautenbach. Linear algebraic techniques for place/transition nets. In W. Brauer, W. Reisig, and G Rozenberg, editors, *Petri Nets: Central Models and their Properties. Advances in Petri Nets 1986*, volume 254 of *Lecture Notes in Computer Science*, pages 142–167. Springer-Verlag, 1987.

Irina A. Lomazova. Nested Petri nets – a formalism for specification of multi-agent distributed systems. *Fundamenta Informaticae*, 43(1-4):195–214, 2000.

Irina A. Lomazova and Philippe Schnoebelen. Some decidability results for nested Petri nets. In *International Conference on Perspectives of System Informatics (PSI'99)*, volume 1755 of *Lecture Notes in Computer Science*, pages 208–220. Springer-Verlag, 2000.

Irina A. Lomazova, Kees M. van Hee, Olivia Oanea, Alexander Serebrenik, Natalia Sidorova, and Marc Voorhoeve. Nested nets for adaptive systems. In *Application and Theory of Petri Nets and Other Models of Concurrency*, Lecture Notes in Computer Science, pages 241–260. Springer-Verlag, 2006.

Ernst W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal Computation*, 13(3):441–460, August 1984.

Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, parts 1-2. *Information and computation*, 100(1):1–77, 1992.

James L. Peterson and Abraham Silberschatz. *Operating System Concepts*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1985. Second edition.

J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall Inc., Englewood Cliffs NJ, 1981.

Carl Adam Petri. Introduction to general net theory. In W. Brauer, editor, *Net Theory and its applications. Proceedings of the Advanced course on general net theory of processes and systems*, volume 84 of *Lecture Notes in Computer Science*. Springer-Verlag, 1979.

Manuel Silva, Enrique Teruel, and José Manuel Colom. Linear algebraic and linear programming techniques for the analysis of place/transition net systems. In W. Reisig and G. Rozenberg, editors, *Lecture Notes in Computer Science: Lectures on Petri Nets I: Basic Models*, volume 1491 of *Advances in Petri Nets*, pages 309–373. Springer-Verlag, 1998.

Rüdiger Valk. Modelling concurrency by task/flow EN systems. In *3rd Workshop on Concurrency and Compositionality*, number 191 in GMD-Studien, St. Augustin, Bonn, 1991. Gesellschaft für Mathematik und Datenverarbeitung.

Rüdiger Valk. Petri nets as token objects: An introduction to elementary object nets. In Jörg Desel and Manuel Silva, editors, *Application and Theory of Petri Nets*, volume 1420 of *Lecture Notes in Computer Science*, pages 1–25, 1998.

Rüdiger Valk. Object Petri nets: Using the nets-within-nets paradigm. In Jörg Desel, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Advanced Course on Petri Nets 2003*, volume 3098 of *Lecture Notes in Computer Science*, pages 819–848. Springer-Verlag, 2003.

Fernando Rosa Velardo and David de Frutos-Escrig. Name creation vs. replication in petri net systems. *Fundam. Inform.*, 88(3):329–356, 2008.

Jan Vitek and Giuseppe Castagna. Seal: A framework for secure mobile computations. In *ICCL Workshop: Internet Programming Languages*, pages 47–77, 1998.

Dianxiang Xu and Yi Deng. Modeling mobile agent systems with high level Petri nets. In *IEEE International Conference on Systems, Man, and Cybernetics'2000*, 2000.