Petri Nets and Higher-Dimensional Automata

Amazigh Amrane[1], Hugo Bazille[1], Uli Fahrenberg[12],
Loïc Hélouët[2], and Philipp Schlehuber-Caissier[3⋆]

[1] EPITA Research Lab (LRE), France
[2] IRISA & Inria Rennes, France
[3] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France

Abstract. Petri nets and their variants are often considered through their interleaved semantics, *i.e.*, considering executions where, at each step, a single transition fires. This is clearly a miss, as Petri nets are a true concurrency model. This paper revisits the semantics of Petri nets as higher-dimensional automata (HDAs) as introduced by van Glabbeek, which methodically take concurrency into account. We extend the translation to include some common features. We consider nets with inhibitor arcs, under both concurrent semantics used in the literature, and generalized self-modifying nets. Finally, we present a tool that implements our translations.

Keywords: Petri net, Higher-dimensional automaton, Concurrency, Inhibitor arc, Generalized self-modifying net

1 Introduction

We revisit the concurrent semantics of Petri nets as higher-dimensional automata (HDAs). In both Petri nets and HDAs, events may occur simultaneously, and both formalisms make a distinction between parallel composition $a \| b$ and choice $a \cdot b + b \cdot a$. However, Petri nets are often considered through their interleaving semantics, annihilating this difference. As an example, Fig. 1 shows two Petri nets and their HDA semantics (see below for definitions): on the left, transitions a and

Fig. 1. Petri nets and HDAs for interleaving (left) and true concurrency (right).

b are mutually exclusive and may be executed in any order but not concurrently; on the right, there is true concurrency between a and b, signified by the filled-in square of the HDA semantics. In interleaving semantics, no distinction is made between the two nets and both give rise to the transition system on the left.

The relations between Petri nets and HDAs were first explored by van Glabbeek in [31], where an HDA is defined as a labeled precubical set whose cells are hypercubes of different dimensions. More recently, [13] introduces an event-based setting for HDAs, defining their cells as totally ordered sets of labeled events. This framework has led to a number of new developments in the theory of HDAs [4, 5, 14, 16], so here we set out to update van Glabbeek's translation to this event-based setting.

Petri nets are a powerful model that can represent infinite systems, and yet preserve decidability of reachability [25] and coverability [23]. Despite their expressiveness, Petri nets miss some features that are essential to represent program executions. In [17], the authors introduce inhibitor arcs, which allow preventing a transition t from firing when a place connected to t by an inhibitor arc is not empty. Obviously, this construction allows for the implementation of a zero test, which makes Petri nets with inhibitor arcs Turing powerful.

We investigate the concurrent semantics of Petri nets with inhibitor arcs, showing that the a-posteriori semantics of [21] gives again rise to HDAs. For the more liberal a-priori semantics (see again [21]) however, we need to introduce *partial* HDAs in which some cells may be missing, mimicking the fact that some serialisations of concurrent executions are now forbidden. We further expand our work to the generalized self-modifying nets of [11], giving their concurrent semantics as *ST-automata* which themselves generalize partial HDAs.

We have developed a prototype tool which implements the translations from Petri nets to HDAs and from PNIs to partial HDAs.[4] Our implementation is able to deal with standard, weighted and inhibitor arcs in a modular fashion.

This article is organised as follows. We begin in Sections 2 and 3 by recalling HDAs and Petri nets, focusing on their concurrent semantics which allows several transitions to fire concurrently. The following sections present our proper contributions. In Sect. 4, we introduce our translation from Petri nets to HDAs, based on [31]. To overcome the symmetry of the HDAs thus built, Sect. 5 introduces an event order which avoids a factorial blow-up in the construction. We also give several examples to illustrate finer points in HDA semantics.

Then we consider Petri nets with inhibitor arcs in Sect. 6, both under a-posteriori and a-priori semantics, and generalized self-modifying nets in Sect. 7. Section 8 presents our implementation.

2   Higher-Dimensional Automata

Higher-dimensional automata (HDAs) extend finite automata with extra structure which permits to specify independence or concurrency of events. They consist of *cells* which are connected by *face maps*. Each cell has a list of events which are

---

[4] See https://gitlabev.imtbs-tsp.eu/philipp.schlehuber-caissier/pn2hda.

active, and face maps permit to pass from a cell to another in which some events have not yet started or are terminated.

We make this precise. A *conclist* (*concurrency list*) over an alphabet Σ is a tuple $(U, \dashrightarrow, \lambda)$, consisting of a finite set U (of events), a strict total order $\dashrightarrow \subseteq U \times U$ (called the *event order*), and a labeling $\lambda : U \to \Sigma$. Conclists represent labeled events running in parallel. If no confusion may arise, we will often refer to conclists by their underlying set only, writing U instead of $(U, \dashrightarrow, \lambda)$, and do the same for other algebraic structures defined throughout. Let $\square = \square(\Sigma)$ denote the set of conclists over Σ.

Conclists $(U_1, \dashrightarrow_1, \lambda_1)$ and $(U_2, \dashrightarrow_2, \lambda_2)$ are *isomorphic* if there is a mapping $\phi : U_1 \to U_2$ such that $a \dashrightarrow_1 b$ iff $\phi(a) \dashrightarrow_2 \phi(b)$ and $\lambda_2 \circ \phi = \lambda_1$. Isomorphisms between conclists are unique [13], so we may switch freely between conclists and their isomorphism classes and will do so in the sequel without mention.

Remark 1. The event order $\dashrightarrow$ is important as a book-keeping device but otherwise carries no computational meaning (see also Sect. 5 below). It plays a key role in distinguishing between events with the same label (autoconcurrency) and is needed for uniqueness of conclist isomorphisms. Conclists without event order are simply multisets, so conclists are multisets totally ordered by the event order, hence lists or words of Σ^*; but we often write them vertically to emphasize that the elements are running in parallel. Event order goes downwards if not indicated.

A *precubical set*

$$(X, \mathsf{ev}, \{\delta_{A,B;U} \mid U \in \square, A, B \subseteq U, A \cap B = \emptyset\})$$

consists of a set of *cells* X together with a function $\mathsf{ev} : X \to \square$. For a conclist U we write $X[U] = \{x \in X \mid \mathsf{ev}(x) = U\}$ for the cells of type U. Further, for every $U \in \square$ and $A, B \subseteq U$ with $A \cap B = \emptyset$ there are *face maps* $\delta_{A,B;U} : X[U] \to X[U \setminus (A \cup B)]$ which satisfy

$$\delta_{C,D;U \setminus (A \cup B)} \delta_{A,B;U} = \delta_{A \cup C, B \cup D;U} \tag{1}$$

for every $U \in \square$, $A, B \subseteq U$, and $C, D \subseteq U \setminus (A \cup B)$.

We will omit the extra subscript "U" in the face maps and further write $\delta_A^0 = \delta_{A, \emptyset}$ and $\delta_B^1 = \delta_{\emptyset, B}$. The *upper* face maps δ_B^1 transform a cell x into one in which the events in B have terminated; the *lower* face maps δ_A^0 transform x into a cell where the events in A have not yet started. Every face map $\delta_{A,B}$ can be written as a composition
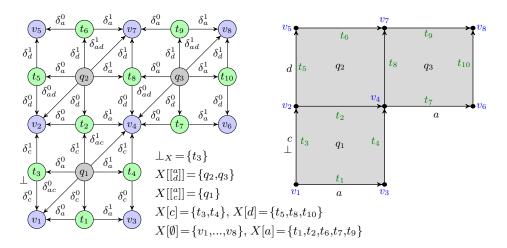
$$\delta_{A,B} = \delta_A^0 \delta_B^1 = \delta_B^1 \delta_A^0,$$

and the *precubical identity* (1) expresses that these transformations commute.

We write $X_n = \{x \in X \mid |\mathsf{ev}(x)| = n\}$ for $n \in \mathbb{N}$ and call elements of X_n *n-cells*. The *dimension* of $x \in X$ is $\dim(x) = |\mathsf{ev}(x)| \in \mathbb{N}$; the dimension of X is $\dim(X) = \sup\{\dim(x) \mid x \in X\} \in \mathbb{N} \cup \{\infty\}$. For $k \in \mathbb{N}$, the *k-truncation* of X is the precubical set $X^{\leq k} = \{x \in X \mid \dim(x) \leq k\}$ with all cells of dimension higher than k removed.

A *higher-dimensional automaton* (*HDA*) $A = (\Sigma, X, \perp)$ consists of a finite alphabet Σ, a precubical set X on Σ, and a subset $\perp \subseteq X$ of initial cells. (We will

Fig. 2. A two-dimensional HDA X on $\Sigma = \{a,c,d\}$, see Ex. 2.

not need accepting cells in this work.) An HDA may be finite or infinite, or even infinite-dimensional.

Computations of HDAs are *paths*, *i.e.*,sequences

$$\alpha = (x_0, \phi_1, x_1, ..., x_{n-1}, \phi_n, x_n) \tag{2}$$

consisting of cells x_i of X and symbols ϕ_i which indicate which type of step is taken: for every $i \in \{1,...,n\}$, (x_{i-1}, ϕ_i, x_i) is either

- $(\delta_A^0(x_i), \nearrow^A, x_i)$ for $A \subseteq \mathsf{ev}(x_i)$ (an *upstep*)
- or $(x_{i-1}, \searrow_A, \delta_A^1(x_{i-1}))$ for $A \subseteq \mathsf{ev}(x_{i-1})$ (a *downstep*).

Intuitively, a downstep terminates events in a cell, following an upper face map. This is why downsteps require that $A \subseteq \mathsf{ev}(x_{i-1})$, *i.e.*, events that are terminated belong to the cell. Similarly, an upstep starts events by following inverses of lower face maps. The constraints on upsteps require that $A \subseteq \mathsf{ev}(x_i)$, *i.e.*, the initiated events belong to the next cell after the step. Both types of steps may be empty.

A cell $x \in X$ is *reachable* if there exists a path α from an initial cell to x, *i.e.*,$x_0 \in \perp$ and $x_n = x$ in the notation (2) above. The *essential* part of X is the subset $\mathsf{ess}(X) \subseteq X$ containing only reachable cells. It is not necessarily an HDA, as some faces may be missing.

Example 2. Figure 2 shows a two-dimensional HDA as a combinatorial object (left) and in a geometric realisation (right). It consists of 21 cells: states $X_0 = \{v_1,...,v_8\}$ in which no event is active ($\mathsf{ev}(v_i) = \emptyset$); transitions $X_1 = \{t_1,...,t_{10}\}$ in which one event is active (*e.g.*,$\mathsf{ev}(t_3) = \mathsf{ev}(t_4) = c$); and squares $X_2 = \{q_1,q_2,q_3\}$ with $\mathsf{ev}(q_1) = [^a_c]$ and $\mathsf{ev}(q_2) = \mathsf{ev}(q_3) = [^a_d]$.

The arrows between cells in the left representation correspond to the face maps connecting them. For example, the upper face map δ_{ac}^1 maps q_1 to v_4 because the

latter is the cell in which the active events a and c of q_1 have been terminated. On the right, face maps are used to glue cells, so that for example $\delta^1_{ac}(q_1)$ is glued to the top right of q_1. In this and other geometric realisations, when we have two concurrent events a and c with $a \dashrightarrow c$, we will draw a horizontally and c vertically.

The HDA X of Fig. 2 admits several paths, for example $t_3 \nearrow^a q_1 \searrow_c t_2 \nearrow^d$ $q_2 \searrow_a t_8 \nearrow^a q_3 \searrow_{ad} v_8$. Note that $\mathsf{ess}(X) = X \setminus \{v_1, t_1, v_3\}$ is not an HDA, and $X = X^{\leq k}$ for all $k \geq 2$.

Remark 3. We often abuse notation and denote conclists by their labels instead of their events, writing for example $\left[\begin{smallmatrix}a\\c\end{smallmatrix}\right]$ for the conclist $(\{e_1 \dashrightarrow e_2\}, \lambda : (e_1 \mapsto a, e_2 \mapsto c))$. (We have already done so in the example above.) As long as there is no autoconcurrency, this abuse of notation is safe and brings no ambiguity; but if we need to assign the same label to several events, we will make events and their labeling explicit in our representation of conclists, writing for instance $\left[\begin{smallmatrix}e_1 \mapsto a\\e_2 \mapsto c\end{smallmatrix}\right]$ for the above example.

3   Petri Nets

A *Petri net* $N = (S, T, F)$ consists of a set of places S, a set of transitions T, with $S \cap T = \emptyset$, and a weighted flow relation $F : S \times T \cup T \times S \to \mathbb{N}$. A *marking* of N is a function $m : S \to \mathbb{N}$. Such marking is *k-bounded* if $m(s) \leq k$ for every place s. It is *bounded* if there is a value $k \in \mathbb{N}$ such that m is k-bounded.

Let X be any set. A function $f : X \to \mathbb{N}$ is a *multiset*, *i.e.*,an extension of sets allowing several instances of each element of X. We introduce some notation for these. We write $x \in f$ if $f(x) \geq 1$. Given two multisets f_1, f_2 over X we will write $f_1 \leq f_2$ iff $f_1(x) \leq f_2(x)$ for every element $x \in X$. If $f(x) \in \{0, 1\}$ for all x, then f may be seen as a set, and the notation $x \in f$ agrees with the usual one for sets. The multisets we use will generally be finite in the sense that $\sum_{x \in X} f(x) < \infty$, and in that case we might use additive notation and write $f = \sum_{x \in X} f(x)x$. This notation easily applies to markings of Petri nets, and we will write for instance $m = 2p_1 + p_4$ for a marking such that $m(p_1) = 2, m(p_4) = 1$, and $m(p_i) = 0$ for any other place $p_i \in S \setminus \{p_1, p_4\}$.

For a transition $t \in T$, the *preset* of t is the multiset ${}^\bullet t : S \to \mathbb{N}$ given by ${}^\bullet t(s) = F(s, t)$. This preset describes how many tokens are consumed in each place when t fires. The *postset* of t is the multiset $t^\bullet : S \to \mathbb{N}$ such that $t^\bullet(s) = F(t, s)$. It describes how many tokens are produced in each place of the net when firing t.
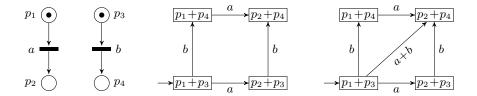
Petri nets compute by transforming markings. Their standard semantics is an interleaved semantics, where states are markings and a single transition can fire at each step. Let $m : S \to \mathbb{N}$ be a marking and $t \in T$, then t can *fire* in m if ${}^\bullet t \leq m$. Firing t produces a new marking $m' = m - {}^\bullet t + t^\bullet$.

The *reachability graph* (see for example [8]) of Petri net $N = (S, T, F)$ is the labeled graph $[\![N]\!]_1 = (V, E)$ given by $V = \mathbb{N}^S$ and

$$E = \{(m, t, m') \in V \times T \times V \mid {}^\bullet t \leq m, m' = m - {}^\bullet t + t^\bullet\}.$$

(The reason for the subscript 1 in $[\![N]\!]_1$ will become clear later.)

In a reachability graph vertices are markings and edges are labeled by the transition which fires. A computation of a Petri net is a path in its reachability graph.

Fig. 3. A Petri net N (left); the reachability graph $[\![N]\!]_1$ (middle); and its concurrent step reachability graph $[\![N]\!]_{\mathrm{CS}}$(right).

Note that we use *collective token semantics*, *i.e.*, tokens in ${}^\bullet t$ that are consumed by firing t are considered as blind resources. Petri nets also have an *individual token* semantics [19] where transitions distinguish tokens individually by considering their origin. This may be used to model realisation of independent processes; but we will not consider it here.

Let N_1, N_2 be two Petri nets. The reachability graphs $[\![N_1]\!]_1 = (V_1, E_1)$ and $[\![N_2]\!]_1 = (V_2, E_2)$ are *isomorphic*, denoted $[\![N_1]\!]_1 \cong [\![N_2]\!]_1$, if there exist bijections $f : V_1 \to V_2$ and $g : E_1 \to E_2$ such that for all $e_1 = (m_1, t_1, m_1') \in E_1$, $g(e_1) = (m_2, t_2, m_2')$ iff $f(m_1) = m_2$ and $f(m_1') = m_2'$.
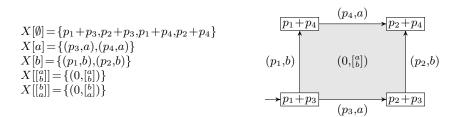
Considering Petri nets via their interleaved semantics misses an important point of the model, namely concurrency. Indeed, it does not allow to distinguish between behaviors where a pair of transitions fire in sequence from behaviors where these transitions are independent and can fire concurrently. One way to cope with this issue is to consider executions of Petri nets as *processes* [19], that is, partial orders representing causal dependencies among transitions occurrences. Another possibility is the use of a *concurrent step semantics* [18], where several transitions are allowed to fire concurrently. The concurrent step semantics mimics that of the interleaved semantics, but fires multisets of transitions.

For a multiset $U : T \to \mathbb{N}$ of transitions we write ${}^\bullet U = \sum_{t \in T} {}^\bullet t\, U(t)$ and $U^\bullet = \sum_{t \in T} t^\bullet U(t)$. U is *firable* in marking m if ${}^\bullet U \leq m$. The *concurrent step reachability graph* [27] of Petri net $N = (S, T, F)$ is the labeled graph $[\![N]\!]_{\mathrm{CS}} = (V, E)$ given by $V = \mathbb{N}^S$ and

$$E = \{(m, U, m') \in V \times \mathbb{N}^T \times V \mid U \neq \emptyset, {}^\bullet U \leq m, m' = m - {}^\bullet U + U^\bullet\}. \qquad (3)$$

Figure 3 shows a simple example of a Petri net and its two types of reachability graph. Note that transitions in $[\![N]\!]_{\mathrm{CS}}$ allow multisets of transition rather than only sets, thus several occurrences of a transition may fire in a concurrent step. This feature is called autoconcurrency, and it is well known that allowing autoconcurrency increases the expressive power of Petri nets [30]. Further, $[\![N]\!]_{\mathrm{CS}}$ is *closed under substeps* in the sense that for all multisets $V \subseteq U$, if $(m, U, m'') \in E$, then we also have $(m, V, m') \in E$ and $(m', U \setminus V, m'') \in E$ for some marking m'.

Notice that our definition of Petri nets allows preset-free transitions t with ${}^\bullet t = \emptyset$. When a transition t has an empty preset, then t is firable from any marking. In an interleaved semantics, allowing preset-free transitions does not change expressive power, so one frequently assumes that ${}^\bullet t \neq \emptyset$ for every $t \in T$. In the

$$X[\emptyset] = \{p_1 + p_3, p_2 + p_3, p_1 + p_4, p_2 + p_4\}$$
$$X[a] = \{(p_3, a), (p_4, a)\}$$
$$X[b] = \{(p_1, b), (p_2, b)\}$$
$$X[[\begin{smallmatrix} a \\ b \end{smallmatrix}]] = \{(0, [\begin{smallmatrix} a \\ b \end{smallmatrix}])\}$$
$$X[[\begin{smallmatrix} b \\ a \end{smallmatrix}]] = \{(0, [\begin{smallmatrix} b \\ a \end{smallmatrix}])\}$$

Fig. 4. Higher-dimensional automaton (reachable part only) for the Petri net of Fig. 3. Left: sets of cells; right: geometric realisation (not showing $X[[\begin{smallmatrix} b \\ a \end{smallmatrix}]]$).

setting of a concurrent semantics with autoconcurrency, an arbitrary number of occurrences of each preset-free transitions may fire from any marking, making the transition relation of (3) of infinite degree. We will generally allow preset-free transitions in what follows.

4   From Petri Nets to HDAs

We expand the notion of reachability graph to a higher-dimensional automaton (HDA). The construction is an adaptation of [31, Def. 9] to the event-based setting of HDAs introduced in [13].

Let $N = (S, T, F)$ be a Petri net. Let $\square = \square(T)$ and define $X = \mathbb{N}^S \times \square$ and $\mathsf{ev} : X \to \square$ by $\mathsf{ev}(m, \tau) = \tau$. For $x = (m, \tau) \in X[\tau]$ with $\tau = (t_1, ..., t_n)$ non-empty and $i \in \{1, ..., n\}$, define

$$\delta^0_{t_i}(x) = (m + {}^\bullet t_i, (t_1, ..., t_{i-1}, t_{i+1}, ..., t_n)),$$
$$\delta^1_{t_i}(x) = (m + t_i^\bullet, (t_1, ..., t_{i-1}, t_{i+1}, ..., t_n)).$$

Using (1) to generate the other face maps, this defines a precubical set $[\![N]\!] = X$.

The 0-cells in X are markings of N, and in an n-cell of X, n transitions of N are running concurrently: the events of an n-cell (m, τ) are the elements of the conclist $\tau = (t_1, ..., t_n)$ of transitions. If a transition t appears multiple times in this sequence, then it is autoconcurrent. Not all precubical sets are in the image of the translation from Petri nets, see [31, Fig. 11] for an example.

Note that we translate (unlabeled) Petri nets to HDAs with *labeled* events: the labels of the events in $[\![N]\!]$ are the transitions of N. A path in $[\![N]\!]$ is a computation in N in which concurrent transitions can fire concurrently.

Lemma 4. *The reachability graph of Petri net N is isomorphic to the 1-truncation of $[\![N]\!]$: $[\![N]\!]_1 \cong [\![N]\!]^{\leq 1}$.*

In order for this statement to make sense, we must consider $[\![N]\!]^{\leq 1}$ as a graph: vertices are 0-cells of $[\![N]\!]$, and edges triples of the form $(\delta^0_t(x), x = (m, t), \delta^1_t(x))$.

We can also relate $[\![N]\!]$ to its concurrent step reachability graph, as follows. For a sequence $a = (a_1, ..., a_n) \in \square(\Sigma)$ on some alphabet Σ denote by $\mathsf{pi}(a) : \Sigma \to \mathbb{N}$ its

Parikh image, *i.e.*, the multiset given by counting symbols: $\mathrm{pi}(a)(x)=|\{i\,|\,a_i=x\}|$. For a precubical set X on Σ define a labeled graph $\mathsf{flat}(X)=(V,E)$ (the *flattening of X*) by $V=X_0$ and $E\subseteq V\times\mathbb{N}^{\Sigma}\times V$ given by

$$E=\{(x,U,z)\,|\,\exists y\in X:\delta^0_{\mathsf{ev}(y)}(y)=x,\delta^1_{\mathsf{ev}(y)}(y)=z,\mathrm{pi}(\mathsf{ev}(y))=U\}.$$

That is, edges in $\mathsf{flat}(X)$ are labeled by multisets of events for which there exist corresponding cells in X, identifying all permutations in one edge. We may now reformulate Lem. 4 above to $[\![N]\!]_1\cong\mathsf{flat}([\![N]\!]^{\leq 1})$, and for $[\![N]\!]_{\mathrm{CS}}$ the following is clear.

Lemma 5. *The concurrent step reachability graph of a Petri net N is isomorphic to the flattening of $[\![N]\!]$: $[\![N]\!]_{\mathrm{CS}}\cong\mathsf{flat}([\![N]\!])$.*   □

Note that under this translation, $[\![N]\!]_{\mathrm{CS}}$ being closed under substeps corresponds to the fact that in $[\![N]\!]$, all faces of any cell are also present.

A Petri net $N=(S,T,F)$ together with an initial marking $i:S\to\mathbb{N}$ is called a *marked Petri net*. Now $i\in[\![N]\!][\emptyset]$, so this induces an HDA $[\![N]\!]=(T,X,\perp)$ with $\perp=\{i\}$.

A marked net N is *bounded* if all markings reachable from i in $[\![N]\!]_1$ are $k-$bounded for some $k\in\mathbb{N}$. Obviously, as firable transitions only depend on the current marking, and as the effect of a firing is deterministic, when a marked net is bounded, the reachable part of $[\![N]\!]_1$, *i.e.*, $\mathsf{ess}([\![N]\!]^{\leq 1})$ is finite. However, due to autoconcurrency, this property does not hold for the full $\mathsf{ess}([\![N]\!])$, as show in the following example.

Example 6. Let $N=(\emptyset,\{a\},F)$ be a Petri net with a single transition a, without places, and with an empty flow relation. With empty initial marking, N is bounded. Now $^\bullet a=\emptyset$, so a is firable in arbitrary autoconcurrency. In $\mathsf{ess}([\![N]\!])$ we get one cell in *every* dimension n: the n-fold autoconcurrency of a. That is, $\mathsf{ess}([\![N]\!])$ is infinite-dimensional (and hence infinite).

Proposition 7. *If marked Petri net N is bounded and has no preset-free transitions, then $\mathsf{ess}([\![N]\!])$ is finite.*

Figure 4 shows the HDA $\mathsf{ess}([\![N]\!])$ for the Petri net N of Fig. 3 with initial marking $i=p_1+p_3$. In particular, since by construction 0-cells in X are markings of N, $\mathsf{ess}([\![N]\!])$ includes all faces, so is an HDA. (This is a general principle: for any HDA X with $\perp_X\subseteq X_0$, $\mathsf{ess}(X)$ is also an HDA [16].)

Note that the 2-dimensional cell $(0,[^a_b])$ corresponds to the edge of $[\![N]\!]_{\mathrm{CS}}$ between p_1+p_3 and p_2+p_4 in Fig. 3. Actually, we get *two* 2-dimensional cells, one with event $[^a_b]$ and the other with $[^b_a]$. (We have omitted the second in the geometric realisation.) This is somewhat unfortunate, as they should denote the same concurrent step $\{a,b\}$. We will show in Sect. 5 how to fix this problem.

5   Event Order

The above definition of the HDA $[\![N]\!]$ is highly symmetric: for a given marking m there is a cell (m,τ) for every sequence $\tau=(t_1,...,t_n)$, even though *in fine* we are

$$X[\emptyset] = \{p_1+p_3+p_5, p_2+p_3+p_5,$$
$$\quad p_1+p_4+p_5, p_2+p_4+p_5\}$$
$$X[a] = \{(p_3,a),(p_4,a)\}$$
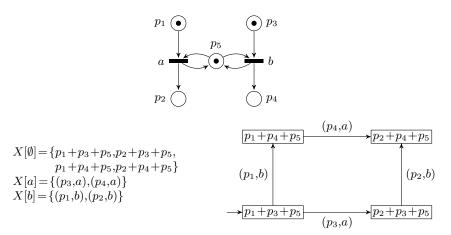$$X[b] = \{(p_1,b),(p_2,b)\}$$

Fig. 5. Petri net (top) and HDA semantics (bottom) of Ex. 8.

only interested in the *multiset* of concurrently active transitions. More precisely, for every permutation $\sigma \in \mathcal{S}_n$ in the nth symmetric group[5] there is a cell $(m,\tau \circ \sigma)$.

That is, $X = [\![N]\!]$ is a *symmetric* precubical set [20, 28]: a precubical set equipped with actions $X_n \times \mathcal{S}_n \to X_n$ of the symmetric groups which are consistent with the face maps, see [20, Sect. 6].

In order to avoid this factorial blow-up, we may fix an arbitrary (non-strict) total order $\preccurlyeq$ on the transitions in T and then instead of $\square(T) \cong T^*$ consider the set

$$T^*_{\preccurlyeq} = \{(t_1,...,t_n) \,|\, \forall i=1,...,n-1 : t_i \preccurlyeq t_{i+1}\}.$$

The definition of the face maps of this reduced $X = [\![N]\!]$ stays the same, and X is now a (non-symmetric) precubical set with one cell for every marking m and every *multiset* of transitions τ.

The order $\preccurlyeq$ on T may be chosen arbitrarily, and changing it amounts to applying a permutation on X and passing to another, equivalent version of (the symmetrization of) X. Technically speaking, the forgetful functor from symmetric precubical sets to precubical sets is a *geometric morphism* [12,24] in that it has both a left and a right adjoint; this is precisely what is needed to be able to say that the order $\preccurlyeq$ on T is arbitrary and may be chosen and re-chosen at will.

We give some further examples of Petri nets and their HDA semantics, using a lexicographic order on transitions.

Example 8. Figure 5 shows a Petri net N which executes a and b in mutual exclusion. We again show the essential part $X = \mathsf{ess}([\![N]\!])$; the initial cell is $p_1+p_3+p_5$. We prove that $X[[^a_b]] = \emptyset$, so that $[\![N]\!]$ is in fact isomorphic to the reachability graph $[\![N]\!]_1$. Assume $x = (m,[^a_b]) \in X[[^a_b]]$, then $\delta^0_{ab}(x) = m+p_1+p_3+2p_5$, but there is no reachable marking m' with $m'(p_5) = 2$.

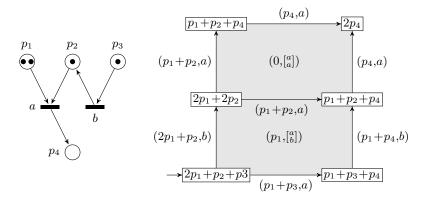[5] The group of permutations of $\{1,...,n\}$.

Fig. 6. Petri net (left) and HDA semantics (right) of Ex. 9. Cells with the same label are identified.

Example 9. Figure 6 shows another Petri net and its HDA semantics. Note that there is *contact* between the transitions a and b: the pre-place p_2 is modified when firing b. Firing b before a enables autoconcurrency of a. The illustration of the HDA is not quite correct, as the autoconcurrent square $x = (0, \begin{bmatrix} e_1 \mapsto a \\ e_2 \mapsto a \end{bmatrix})$ only has *two* different faces:[6] using $\cong$ for conclist isomorphism, we have $\delta^0_{e_1}(x) = (p_1 + p_2, e_2) \cong \delta^0_{e_2}(x) = (p_1 + p_2, e_1)$ and $\delta^1_{e_1}(x) = (p_4, e_2) \cong \delta^1_{e_2}(x) = (p_4, e_1)$.

Example 10. Figure 7 shows a slightly more complicated example, where transitions a and b initially are mutually exclusive, but then c introduces independence. Geometrically this is an empty box without bottom face ("Fahrenberg's matchbox" [10]). We have $X\left[\begin{bmatrix} a \\ b \\ c \end{bmatrix}\right] = \emptyset$, for if $x = \left(m, \begin{bmatrix} a \\ b \\ c \end{bmatrix}\right) \in X\left[\begin{bmatrix} a \\ b \\ c \end{bmatrix}\right]$, then $\delta^0_{abc}(x) = m + p_1 + p_3 + 2p_5 + p_6$, which is unreachable.

6   Inhibitor Arcs

We now extend our setting to Petri nets with inhibitor arcs. A *Petri net with inhibitor arcs* (*PNI*) $N = (S,T,F,I)$ consists of a Petri net (S,T,F) and a set $I \subseteq S \times T$ of inhibitor arcs. We denote by $^\circ t = \{s \in S \mid (s,t) \in I\}$ the *inhibitor places* of $t \in T$.

The interleaved semantics for PNIs is as follows. Tokens in inhibitor places keep transitions from being firable, so a transition $t \in T$ can fire in marking $m : S \to \mathbb{N}$ if $^\bullet t \leq m$ and $\forall s \in {}^\circ t : m(s) = 0$. The reachability graph of a PNI $N = (S,T,F,I)$ is the labeled graph $[\![N]\!]_1 = (V,E)$ given by $V = \mathbb{N}^S$ and $E \subseteq V \times T \times V$ with

$$E = \{(m,t,m') \mid {}^\bullet t \leq m, \forall s \in {}^\circ t : m(s) = 0, m' = m - {}^\bullet t + t^\bullet\}.$$

PNIs have two different concurrent semantics, one which disables concurrent steps in which transitions may inhibit each other and one which does not. These are called, respectively, *a-posteriori* and *a-priori* semantics in [21], and we treat them both below. We refer to [21, Sect. 2] for an in-depth discussion of these semantics.

---

[6] We need to use the extended conclist notation here due to autoconcurrency.

$$X[\emptyset] = \{A,B,C,D,E,F,G,H\}$$
$$A = p_1 + p_3 + p_5 + p_6$$
$$B = p_2 + p_3 + p_5 + p_6$$
$$C = p_1 + p_4 + p_5 + p_6$$
$$D = p_2 + p_4 + p_5 + p_6$$
$$E = p_1 + p_3 + 2p_5 \qquad F = p_2 + p_3 + 2p_5$$
$$G = p_1 + p_4 + 2p_5 \qquad F = p_2 + p_4 + 2p_5$$
$$X[a] = \{(p_3 + p_5, a), (p_4 + p_5, a),$$
$$(p_3 + p_6, a), (p_4 + p_6, a)\}$$
$$X[b] = \{(p_1 + p_5, b), (p_2 + p_5, b),$$
$$(p_1 + p_6, b), (p_2 + p_6, b)\}$$
$$X[c] = \{(p_1 + p_3 + p_5, c), (p_2 + p_3 + p_5, c),$$
$$(p_1 + p_4 + p_5, c), (p_2 + p_4 + p_5, c)\}$$
$$X[[{}^a_b]] = \{(0, [{}^a_b])\}$$
$$X[[{}^a_c]] = \{(p_3, [{}^a_c]), (p_4, [{}^a_c])\}$$
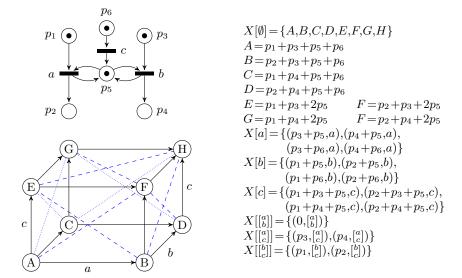$$X[[{}^b_c]] = \{(p_1, [{}^b_c]), (p_2, [{}^b_c])\}$$

Fig. 7. Petri net and HDA semantics of Ex. 10. Concurrent squares are indicated using blue crosses instead of filled.

6.1 Concurrent a-posteriori semantics

In the a-posteriori semantics, a multiset U of transitions is firable in m if

(1) ${}^\bullet U \le m$;
(2) for every $t \in U$ and every place $s \in {}^\circ t$, $m(s) = 0$;
(3) for every $t_1, t_2 \in U$ such that $t_1 = t_2$ implies $U(t_1) \ge 2$, $t_1^\bullet \cap {}^\circ t_2 = \emptyset$.

The last condition ensures that t_1 cannot produce a token that prevents t_2 from firing, even when t_1 and t_2 are autoconcurrent in U. With this condition, the transitions in U cannot inhibit *each other*. One advantage of a-posteriori semantics is that $[\![N]\!]_{CS}$ is closed under substeps [7, Prop. 2.8].

Similarly to what we did before for Petri nets, we can give an HDA semantics to a PNI $N = (S,T,F,I)$ under a-posteriori concurrent semantics, by restricting the cells to satisfy conditions (2) and (3) above. We let again $\square = \square(T)$ and define

$$X = \{(m, (t_1, ..., t_n)) \in \mathbb{N}^S \times \square \mid \forall i = 1, ..., n : \forall s \in {}^\circ t_i : m(s) = 0,$$
$$\forall i \ne j = 1, ..., n : t_i^\bullet \cap {}^\circ t_j = \emptyset\}.$$

The rest of the definition of $[\![N]\!]$ now proceeds as before, and [7, Prop. 2.8] ensures that for any $x \in X$ and any $A \subseteq \mathsf{ev}(x)$, we also have $\delta_A^0(x), \delta_A^1(x) \in X$.

Example 11. Figure 8 shows the Petri net of Fig. 3 with an added inhibitor arc from p_4 to a. That is, transition a is disabled when there is a token in p_4, hence $X[a]$ contains only (p_3, a) but not (p_4, a). Compared to Fig. 4, the 2-cell $(0, [{}^a_b])$ is also disabled, as $b^\bullet = {}^\circ a = \{p_4\}$: there is no concurrency.

$$X[\emptyset] = \{p_1+p_3, p_2+p_3, p_1+p_4, p_2+p_4\}$$
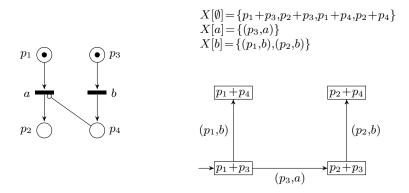$$X[a] = \{(p_3,a)\}$$
$$X[b] = \{(p_1,b),(p_2,b)\}$$

Fig. 8. PNI and HDA semantics of Ex. 11.

[7] also introduces a subclass of PNIs called *primitive systems* defined as follows. For a marked PNI $N = (S,T,F,I,i)$, let $Inib(N) = \{s \in S \mid \exists (s,t) \in I\}$ be the set of places used to inhibit some transition. Now N is called a *primitive system* if there exists a function $EL : Inib(N) \to \mathbb{N}$ such that for all $s \in Inib(N)$ and all reachable markings m with $m(s) > EL(s)$, if m' is reachable from m, then for all $t \in T$ with ${}^\bullet t \le m'$, we have $s \notin {}^\circ t$.

Intuitively, in primitive systems, when the bound $EL(s)$ is exceeded in some reachable marking m, then no transition with $s \in {}^\circ t$ is fired in markings that are reachable from m. [7] demonstrates that primitive systems can be simulated by Petri nets (without inhibitor arcs). However, the author shows that while her construction preserves interleaved semantics, it does not preserve concurrent step semantics; it is clear that the same is true for our HDA semantics.

6.2   Concurrent a-priori semantics

In the more liberal a-priori concurrent semantics, condition (3) of the multiset firing rules is removed. For intuition, consider again Ex. 11 and Fig. 8. In a-posteriori semantics, the concurrent step $U = \{a,b\}$ is disabled due to condition (3): b produces a token in inhibitor place p_4 connected to a. This seems rather restrictive: one might argue that *while* the b transition is firing, it has not yet produced a token in p_4, so it should not prevent from starting the firing of a.

On the other hand, if we add the 2-cell $(0,[{}^a_b])$ to the semantics, we are also forced to add its upper face $\delta^1_b((0,[{}^a_b])) = (p_4,a)$, given that all faces of cells must be present in HDAs. Now the cell (p_4,a) would have the upper left vertex $p_1 + p_4$ as a lower face, so semantically that means that we can fire a after firing b, which is clearly contrary to what an inhibitor arc should do. To give proper semantics to PNIs we thus must allow HDAs in which some faces are "missing". These are called *partial HDAs* and have been introduced in [9,15]; we adapt their definition to our event-based setting.

A *partial precubical set*

$$(X, \mathsf{ev}, \{\delta_{A,B;U} \mid U \in \square, A,B \subseteq U, A \cap B = \emptyset\})$$

consists of a set of cells X together with a function $\mathsf{ev} : X \to \square$. Further, for every $U \in \square$ and $A, B \subseteq U$ with $A \cap B = \emptyset$ there are partial face maps $\delta_{A,B;U} : X[U] \to X[U \setminus (A \cup B)]$ which satisfy

$$\delta_{C,D;U \setminus (A \cup B)} \delta_{A,B;U} \subseteq \delta_{A \cup C, B \cup D;U} \tag{4}$$

for every $U \in \square$, $A, B \subseteq U$, and $C, D \subseteq U \setminus (A \cup B)$. Except for the face maps being partial, this is the same definition as for HDAs in Section 2; we again omit the extra subscripts "U". By the notation $\subseteq$ in (4) we mean that if $\delta_{A,B}$ and $\delta_{C,D}$ are defined, then also $\delta_{A \cup C, B \cup D}$ is defined and equal to the composition $\delta_{C,D} \delta_{A,B}$; but $\delta_{A \cup C, B \cup D}$ may be defined without one or both of the maps on the left-hand side being defined.

A *partial higher-dimensional automaton*, or *pHDA*, $(\Sigma, X, \bot)$ consists of a partial precubical set X on Σ together with a subset $\bot \subseteq X$ of initial cells.

We now give a-priori concurrent semantics to PNIs by translating them to pHDAs. Let $N = (S, T, F, I)$ be a PNI and $X = [\![(S, T, F)]\!]$ its standard HDA semantics, ignoring inhibitor arcs. Define a subset $X' \subseteq X$ by

$$X' = \{(m, (t_1, ..., t_n)) \in \mathbb{N}^S \times \square \mid \forall i = 1, ..., n : \forall s \in {}^\circ t_i : m(s) = 0\}$$

and let $[\![N]\!] = X'$.

Hence a cell (m, τ) exists in $[\![N]\!]$ if none of the tokens in m inhibits any transition in τ. Compared to the contact-free semantics, we leave out Busi's last condition (3), thus allowing firing of subsets containing pairs of transitions t_1, t_2 with $t_1^\bullet \cap {}^\circ t_2 \neq \emptyset$.

Lemma 12. $[\![N]\!]$ *is a partial precubical set.*

If we extend the definition of flattening and truncation to partial HDAs and allow the concurrent step semantics to not be closed under substeps, we have the following analogues of Lemmas 4 and 5.
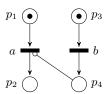
Lemma 13. *The reachability graph of PNI N is isomorphic to the 1-truncation of $[\![N]\!]$: $[\![N]\!]_1 \cong [\![N]\!]^{\leq 1}$.* $\square$

Lemma 14. *The concurrent step reachability graph of PNI N is isomorphic to the flattening of $[\![N]\!]$: $[\![N]\!]_{CS} \cong \mathsf{flat}([\![N]\!])$.* $\square$

We also have the following analogue to Prop. 7.

Proposition 15. *If a marked PNI N is bounded and has no preset-free transitions, then $\mathsf{ess}([\![N]\!])$ is finite.* $\square$

Example 16. Consider Fig. 9, where Ex. 11 is interpreted with partial HDA semantics, ignoring condition (3). The 2-cell $(0, [{}^a_b])$ is now present, but its δ^1_b face is not. There are now two paths from $p_1 + p_3$ to $p_2 + p_4$, passing respectively through $p_2 + p_3$ and $(0, [{}^a_b])$. The latter captures reaching marking $p_2 + p_4$ from $p_1 + p_3$ through the multistep $U = \{a, b\}$ in $[\![N]\!]_{CS}$.
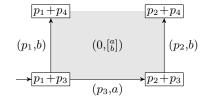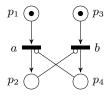
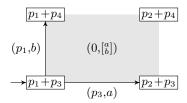**Fig. 9.** PNI and partial HDA semantics of Ex. 16.

Fig. 10. Second PNI and partial HDA semantics of Ex. 16.

We may also modify the example by introducing another inhibitor arc from p_2 to b, see Fig. 10. Then transition a (resp. b) is disabled when there is a token in p_4 (resp. p_2). Again, the corresponding partial HDA contains the 2-cell $(0,[\begin{smallmatrix}a\\b\end{smallmatrix}])$ even if both its δ_a^1 and δ_b^1 are missing. In the a-posteriori semantics, the marking p_2+p_4 is now unreachable; but in the a-priori semantics, there is a path passing through $(0,[\begin{smallmatrix}a\\b\end{smallmatrix}])$ which mimics firing a and b in parallel.

Similarly to [7, Thm. 5.20], it can be proven that the concurrent a-priori semantics of primitive systems may not be simulated by Petri nets without inhibitor arcs.
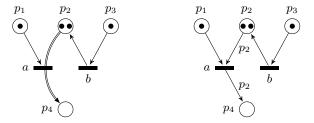
7   Self-Modifying Nets

Instead of considering other extensions one by one, we now pass to generalized self-modifying nets which encompass many other extensions. Recall [11] that a *generalized self-modifying net* (*G-net*) $N = (S,T,F)$ consists of a set of places S, a set of transitions T, with $S \cap \mathbb{N} = S \cap T = \emptyset$, and a weighted flow relation $F : S \times T \cup T \times S \to \mathbb{N}[S]$.

That is, flow arcs are labeled by *polynomials* in place variables; contrary to [11] we do not assume that the labels are sums of monomials. We will propose a concurrent semantics using ST-*graphs*, a generalisation of partial HDAs, see below.

The intuition of the flow polynomials labeling arcs is that when a transition fires, it consumes precisely the number of tokens given by evaluating polynomials of its input arcs, and produces precisely the number of tokens given by evaluating polynomials labeling its output arcs in the current marking. More precisely,

 – if $F(s,t) = P$ for an arc (s,t), then firing t consumes $P(m)$ tokens from s, where m is the current marking; so the polynomial P is evaluated by replacing its place variables with the current number of tokens in the respective places;
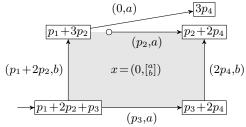
Fig. 11. Petri net with transfer arc (left) and its translation to a G-net (right). Annotations "p_2" at flow arcs indicate that they consume and produce the number of tokens present in p_2. Otherwise, they consume one token as usual.

- if $F(t,s) = P$ for an arc (t,s), then firing t produces $P(m)$ tokens in s, where m is the marking *before* t started firing.

That is, in interleaved semantics the new marking when firing a transition t in marking m is given by $m' = m - {}^\bullet t(m) + t^\bullet(m)$: ${}^\bullet t$ is the function $\phi : S \to \mathbb{N}[S]$ given by $\phi(s) = F(s,t)$, so ${}^\bullet t(m) : S \to \mathbb{N}$ is given by ${}^\bullet t(m)(s) = \phi(s)(m)$. Now in concurrent semantics, other transitions may fire between starting and terminating t, so we will need to remember the marking before firing t, see below.

Example 17. G-nets can model *transfer arcs*, a Petri net extension which, when firing the associated transition, transfers *all* tokens present in the pre-place to the post-place. Figure 11 shows a simple Petri net N with a transfer arc, from p_2 through the a-transition to p_4, and its G-net translation. (In fact, this is a self-modifying net in the sense of [29], a strict subclass of G-nets.) Note that there is contact between the transitions a and b, and the marking $p_2 + 2p_4$ reached by firing first a and then b is not reachable from the marking $p_1 + 3p_3$ obtained after firing b.

The latter is indicated with a "broken" arrow in the intuitive (partial) HDA semantics for N:

Transitions a and b can fire concurrently, so we have a square $x \in [\![N]\!][\![{a \brack b}]\!]$, and from here b can terminate, leading to $\delta_b^1(x) = (p_2, a)$. Since $p_2 + 2p_4$ is not reachable from $p_1 + 3p_2$, we must have $\delta_a^0((p_2, a)) \neq p_1 + 3p_2$ if it is defined. In fact, this is unreachable, so it is not defined. We have a partial HDA, with all cells "geometrically present" but one gluing undefined.

In the previous example and Ex. 21 below, pHDAs allow to capture the concurrent semantics of G-nets, however, we conjecture that they are not sufficient in the general case. Instead, in order to give concurrent semantics to G-nets, we introduce

a third kind of automaton, closely related to HDAs and partial HDAs but more general. First, some terminology; again Σ denotes an arbitrary alphabet and $\square = \square(\Sigma)$. A *starter* is a pair (A, U), written $_A{\uparrow}U$, consisting of a conclist $U \in \square$ and a subset $A \subseteq U$. A *terminator* is a pair (U, B), written $U{\downarrow}_B$, consisting of a conclist U and a subset $B \subseteq U$. The intuition is that these denote actions of starting resp. terminating subsets of the events in U, passing from $U \setminus A$ to U, resp. from U to $U \setminus B$.

Let $\mathrm{ST} = \mathrm{ST}(\Sigma)$ denote the (infinite) set of starters and terminators over Σ. An *ST-graph* is a structure (Σ, Q, E, λ) consisting of a set Q of states, a set $E \subseteq Q \times \mathrm{ST} \times Q$, and a labeling $\lambda : Q \to \square$ such that for all $(p, x, q) \in E$,

- if $x = {}_A{\uparrow}U$, then $\lambda(p) = U \setminus A$ and $\lambda(q) = U$;
- if $x = U{\downarrow}_B$, then $\lambda(p) = U$ and $\lambda(q) = U \setminus B$.

ST-*automata*, *i.e.*, ST-graphs with initial (and final) states, have been introduced in [3] in order to give operational semantics to HDAs.

Now let $N = (S, T, F)$ be a G-net and define an ST-graph $[\![N]\!]'_{\mathrm{ST}} = (T, Q', E', \lambda')$ by $Q' = \mathbb{N}^S \times \square \times (\mathbb{N}^S)^*$, $\lambda'(m, \tau, \mu) = \tau$, and

$$
\begin{aligned}
E' = \Big\{ & \Big(\big(m + {}^\bullet t_i(\mu_i), (t_1, ..., t_{i-1}, t_{i+1}, ..., t_n), (\mu_1, ..., \mu_{i-1}, \mu_{i+1}, ..., \mu_n)\big), \\
& \quad {}_{t_i}{\uparrow}(t_1, ..., t_n), \big(m, (t_1, ..., t_n), \mu\big) \Big) \Big\} \\
\cup \Big\{ & \Big(\big(m, (t_1, ..., t_n), \mu\big), (t_1, ..., t_n){\downarrow}_{t_i}, \\
& \quad \big(m + t_i^\bullet(\mu_i), (t_1, ..., t_{i-1}, t_{i+1}, ..., t_n), (\mu_1, ..., \mu_{i-1}, \mu_{i+1}, ..., \mu_n)\big) \Big) \Big\}.
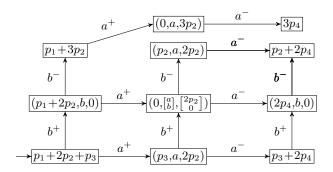\end{aligned}
$$

The intuition is that in a state $x = (m, (t_1, ..., t_n), (\mu_1, ..., \mu_k)) \in Q'$, $n = k$ if x is reachable, and the marking $\mu_i : S \to \mathbb{N}$ is the memory of how the net was marked before transition t_i started firing.

Remark 18. In the construction of $[\![N]\!]'_{\mathrm{ST}}$, one may compose successive starting edges to start multiple transitions at the same time, similarly for terminating edges. (See [3, Sect. 4.4] for a related construction.) Note that concurrency of several transitions is captured by starting them one by one in any order before terminating any of them.

We have primed $[\![N]\!]'_{\mathrm{ST}}$ above, as the memory so-defined remembers too much: in Ex. 17, transition a only needs to remember the contents of p_2 and b should not require memory at all. We remedy this by introducing a notion of memory equivalence and passing to a quotient. Say that two pairs (t, m), (t, m') are *memory equivalent*, denoted $(t, m) \sim (t, m')$, if ${}^\bullet t(m) = {}^\bullet t(m')$ and $t^\bullet(m) = t^\bullet(m')$. Then m and m' have the same *effect* on the net when t is fired.

Now extend $\sim$ to Q' by $\big(m, (t_1, ..., t_n), (m_1, ..., m_n)\big) \sim \big(m, (t_1, ..., t_n), (m'_1, ..., m'_n)\big)$ if $(t_i, m_i) \sim (t_i, m'_i)$ for all i. As the memory works by insertion and deletion when starting resp. terminating transitions, $\sim$ is a congruence on the ST-graph $[\![N]\!]'_{\mathrm{ST}}$ and we may form the quotient $[\![N]\!]_{\mathrm{ST}} = [\![N]\!]'_{\mathrm{ST}} / {\sim}$.

Similarly to HDAs, we may define the 1-truncation of an ST-graph (Σ, Q, E, λ) as the graph which has as vertices states $q \in Q$ for which $\dim(\lambda(q)) = 0$ and edges

$$a^+ \longrightarrow \boxed{(0,a,3p_2)} \xrightarrow{\ a^-\ } \boxed{3p_4}$$

$$\boxed{p_1+3p_2} \qquad \boxed{(p_2,a,2p_2)} \xrightarrow{\ a^-\ } \boxed{p_2+2p_4}$$

$$b^- \qquad\qquad b^- \qquad\qquad b^-$$

$$\boxed{(p_1+2p_2,b,0)} \xrightarrow{\ a^+\ } \boxed{(0,[^a_b],[^{2p_2}_0])} \xrightarrow{\ a^-\ } \boxed{(2p_4,b,0)}$$

$$b^+ \qquad\qquad b^+ \qquad\qquad b^+$$

$$\boxed{p_1+2p_2+p_3} \xrightarrow{\ a^+\ } \boxed{(p_3,a,2p_2)} \xrightarrow{\ a^-\ } \boxed{p_3+2p_4}$$

Fig. 12. ST-automaton semantics (reachable part only) of the G-net of Fig. 11, see Ex. 20. For readability, transitions are labeled with starts (a^+) and terminations (a^-) of actions rather than starters and terminators.

(q,a,q') corresponding to states $x \in Q$ with $\dim(\lambda(x)) = 1$. This yields the following analogue of Lem. 4; the relation to the concurrent step reachability graph is left for future work.

Lemma 19. *The reachability graph of G-net N is isomorphic to the 1-truncation of $[\![N]\!]_{\mathrm{ST}}$.* □

We leave open the question of a characterization of G-nets (or subclasses) by means of partial HDAs. They are sufficient in all our examples. On the other hand, note that the ST-automaton semantics we have given conforms with how the reachable part of the semantics is constructed, by starting and terminating events one at a time. (See also Sect. 8.)

Example 20. We continue Ex. 17 by giving $[\![N]\!]_{\mathrm{ST}}$ for the transfer net N in Fig. 12, indicating memory equivalence classes by representatives. We see that, as expected, the sequence $b^+b^-a^+$ leads to a different state than the other permutations.

We show some of the calculations; for readability we make a distinction between a variable p appearing in a polynomial and the current number of tokens in p, denoting the latter by $|p|$. We also write ${}^\bullet t = \sum_s F(s,t)s$ to denote that t fires by consuming $F(s,t)$ tokens from each place s (and similarly for $t^\bullet$).

We have ${}^\bullet a = F(p_1,a)p_1 + F(p_2,a)p_2 + F(p_3,a)p_3 = p_1 + |p_2|p_2$ and $a^\bullet = |p_2|p_4$. In addition, the marking before only firing a is $\mu = p_1 + 2p_2 + p_3$. Then ${}^\bullet a(\mu) = p_1 + \mu(p_2)p_2 = p_1 + 2p_2$. Note also that for $\mu' = 2p_2$, for example, ${}^\bullet a(\mu) = {}^\bullet a(\mu')$ and $a^\bullet(\mu) = a^\bullet(\mu')$.

More generally, we have ${}^\bullet a(n_1 p_1 + n_2 p_2 + n_3 p_3) = p_1 + n_2 p_2$ for all $n_1, n_2, n_3 \in \mathbb{N}$. Thus $(p_1 + n_2 p_2 + p_3, a^+, (p_3,a,p_1+n_2 p_2)) \in E'$. We also have that $(p_3,a,p_1+n_2 p_2) \sim (p_3,a,n_2 p_2)$. Indeed, since $F(p_1,a)$ and $F(a,p_1)$ are constants, ${}^\bullet a(p_1 + n_2 p_2) = {}^\bullet a(n_2 p_2)$ and $a^\bullet(p_1 + n_2 p_2) = a^\bullet(n_2 p_2)$.

Thus, for the sequence $a^+ b^+ b^-$, $((p_3,a,2p_2), b^+, (0,[^a_b], [^{2p_2}_{p_3}])) \in E'$ since we had only one token in p_3 before firing b, but $(0,[^a_b], [^{2p_2}_{p_3}]) \sim (0,[^a_b], [^{2p_2}_0])$. Finally, $((0,[^a_b], [^{2p_2}_0]), b^-, (p_2,a,2p_2)) \in E'$, leading to a different state than $b^+ b^- a^+$.
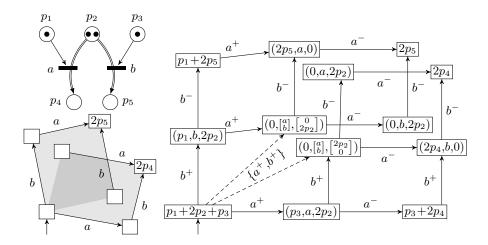
Fig. 13. G-net of Ex. 21 (top left), ST-automaton semantics (right), and corresponding partial HDA (bottom left).

Example 21. Figure 13 shows a G-net containing two transfer arcs which are in conflict and the corresponding ST-automaton semantics. If we admit that starting a and b at the same time is non-deterministic and may lead to any of the two states in the center (as indicated by the dashed $\{a^+, b^+\}$-labeled transitions in ST-automaton), then this is now a partial HDA: two partial squares labeled $\begin{bmatrix} a \\ b \end{bmatrix}$ and glued at the initial state.

8   Implementation

We have developed a prototype tool, `pn2HDA`, written in `C++` and implementing our translations from Petri nets to HDAs and from PNIs to partial HDAs. Our implementation is able to deal with standard, weighted and inhibitor arcs in a modular fashion and is available at `https://gitlabev.imtbs-tsp.eu/philipp.schlehuber-caissier/pn2hda`. Our tool is based on previous work by our student Timothée Fragnaud and on the PNML parser provided by the library Symmetri[7], but any other parser could easily replace this task.

As shown above, Petri nets with or without inhibitor arcs (a-posteriori semantics) can be translated into the same formalism: partial HDAs. The implementation reflects this by having a parametrizable (via templates) representation for Petri nets, which is then used in a generic way to build the corresponding pHDA.

For this prototype tool, we have chosen an explicit representation of the pHDA and its face maps as defined at the beginning of section 4. That is each reachable cell $x \in X$ is defined as the tuple (m, τ) corresponding to the conclist and the marking. As event order, the arbitrary total order on transitions introduced in section 5, we have chosen the shortlex order on transition names (*i.e.*, names are

---

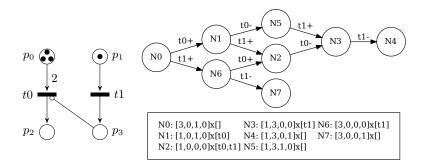[7] See `https://github.com/thorstink/Symmetri`.

Fig. 14. Petri net (left) and pHDA conversion (right): definition of cells (top) and ST-automaton representation (bottom).

first sorted by their length, and sequences of identical length are sorted according to the lexicographical order). While this representation is likely not the most efficient, it allows to underpin the correctness of our constructions.

Since we cannot display the constructed pHDAs if their dimension is greater than 3, we have chosen to output them as ST-automata (see Sect. 7). For gathering information about the structure of the automaton like the number of unique conclists or markings we provide a function `get_csv_data`. In the repository we also provide the PNML files of all Petri nets given as examples in the paper, as well as a selection of models from the Model Checking Contest[8]. This is useful for example for gathering statistics about cells of different dimensions, see Fig. 15.

9   Conclusion

We have seen that Petri nets exhibit a natural concurrent semantics as higher-dimensional automata (HDAs) which allows methodical reasoning about the finer points of the semantics of Petri nets and their extensions. The semantics of Petri nets with inhibitors is naturally expressed using partial HDAs in which some faces may be missing.

We have also given concurrent semantics to generalized self-modifying nets (G-nets) which encompass many other extensions. We have given the semantics as ST-automata, a generalization of partial HDAs, using a notion of memory to store the state of the G-net before the starts of transitions. Whether or not the semantics may also be given as partial HDAs, or whether there are interesting subclasses of G-nets which admit partial-HDA semantics, is left open.

Finally, we have presented an implementation of the translations from Petri nets to HDAs and from Petri nets with inhibitors to partial HDAs.
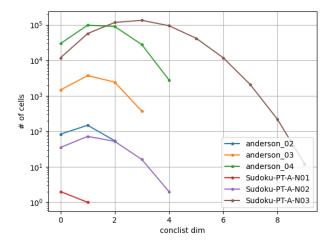
―――――――――
[8] See `https://mcc.lip6.fr`.

Fig. 15. Statistics on cell distribution for some MCC models.

We believe that pHDA and ST-automaton semantics may provide a unifying framework for investigating constructions on Petri nets (such as the removal of inhibitor arcs from primitive systems which we have seen) and their effects on concurrent semantics. This should apply to other well-known simplifications such as removing read arcs; but also for example to unfoldings.

We would also like to apply our setting to other generalizations of Petri nets. Introducing concurrent semantics for affine nets [6], for example, appears difficult; but real-time extensions such as time Petri nets [26] seem natural candidates. Recent work on higher-dimensional timed automata [1,2] will be useful in this context.

Finally, we plan to continue to work on our implementation. We are working on another, implicit, representation of pHDAs which would avoid creating all reachable cells. We would also like to extend our tool to time Petri nets and connect it with Kahl's work on program graphs and homology [22].[9]

Acknowledgements

---

[9] See `https://github.com/twkahl/PG2HDA`.

References

1. Amazigh Amrane, Hugo Bazille, Emily Clement, and Uli Fahrenberg. Languages of higher-dimensional timed automata. In Lars Michael Kristensen and Jan Martijn van der Werf, editors, *PETRI NETS*, volume 14628 of *Lecture Notes in Computer Science*, pages 197–219. Springer, 2024.

2. Amazigh Amrane, Hugo Bazille, Emily Clement, Uli Fahrenberg, and Philipp Schlehuber-Caissier. Higher-dimensional timed automata for real-time concurrency. *CoRR*, abs/2401.17444, 2025.

3. Amazigh Amrane, Hugo Bazille, Emily Clement, Uli Fahrenberg, and Krzysztof Ziemiański. Presenting interval pomsets with interfaces. In Uli Fahrenberg, Wesley Fussner, and Roland Glück, editors, *RAMiCS*, volume 14787 of *Lecture Notes in Computer Science*, pages 28–45. Springer, 2024.

4. Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Marie Fortin. Logic and languages of higher-dimensional automata. In Joel D. Day and Florin Manea, editors, *DLT*, volume 14791 of *Lecture Notes in Computer Science*, pages 51–67. Springer, 2024.

5. Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Krzysztof Ziemiański. Closure and decision properties for higher-dimensional automata. In Erika Ábrahám, Clemens Dubslaff, and Silvia Lizeth Tapia Tarifa, editors, *ICTAC*, volume 14446 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2023.

6. Rémi Bonnet, Alain Finkel, and M. Praveen. Extending the Rackoff technique to affine nets. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *FSTTCS*, volume 18 of *LIPIcs*, pages 301–312. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.

7. Nadia Busi. Analysis issues in Petri nets with inhibitor arcs. *Theoretical Computer Science*, 275(1-2):127–177, 2002.

8. Jörg Desel and Wolfgang Reisig. Place/transition Petri nets. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models: Advances in Petri Nets*, pages 122–173. Springer, 1998.

9. Jérémy Dubut. Trees in partial higher dimensional automata. In Mikołaj Bojańczyk and Alex Simpson, editors, *FOSSACS*, volume 11425 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2019.

10. Jérémy Dubut, Eric Goubault, and Jean Goubault-Larrecq. Natural homology. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP*, volume 9135 of *Lecture Notes in Computer Science*, pages 171–183. Springer, 2015.

11. Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In Kim G. Larsen, Sven Skyum, and Glynn Winskel, editors, *ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1998.

12. Uli Fahrenberg. *Higher-Dimensional Automata from a Topological Viewpoint*. PhD thesis, Aalborg University, Denmark, 2005.

13. Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science*, 31(5):575–613, 2021. `https://arxiv.org/abs/2103.07557`.

14. Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Kleene theorem for higher-dimensional automata. *Logical Methods in Computer Science*, 20(4), 2024.

15. Uli Fahrenberg and Axel Legay. Partial higher-dimensional automata. In Lawrence S. Moss and Pawel Sobocinski, editors, *CALCO*, volume 35 of *Leibniz International Proceedings in Informatics*, pages 101–115. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

16. Uli Fahrenberg and Krzysztof Ziemiański. Myhill-Nerode theorem for higher-dimensional automata. *Fundamentae Informatica*, 192(3-4):219–259, 2024.

17. Michael J. Flynn and Tilak Agerwala. Comments on capabilities, limitations and 'correctness' of Petri nets. *Computer Architecture News*, 4, 1973.

18. Hartmann J. Genrich, Kurt Lautenbach, and Pazhamaneri S. Thiagarajan. Elements of general net theory. In Wilfried Brauer, editor, *Net Theory and Applications*, pages 21–163. Springer, 1980.

19. Ursula Goltz and Wolfgang Reisig. The non-sequential behaviour of Petri nets. *Information and Control*, 57(2):125–147, 1983.

20. Marco Grandis and Luca Mauri. Cubical sets and their site. *Theory and Applications of Categories*, 11(8):185–211, 2003.

21. Ryszard Janicki and Maciej Koutny. Semantics of inhibitor nets. *Information and Computation*, 123(1):1–16, 1995.

22. Thomas Kahl. On the homology language of HDA models of transition systems. *Journal of Applied and Computational Topology*, 8(4):859–873, 2024.

23. Richard M. Karp and Raymond E. Miller. Parallel program schemata. *J. Comput. Syst. Sci.*, 3(2):147–195, 1969.

24. Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic*. Springer, 1992.

25. Ernst W. Mayr. An algorithm for the general Petri net reachability problem. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing, May 11-13, 1981, Milwaukee, Wisconsin, USA*, pages 238–246. ACM, 1981.

26. Philip Merlin. A study of the recoverability of computer systems. *Ph. D. Thesis, Computer Science Dept., University of California*, 1974.

27. Madhavan Mukund. Petri nets and step transition systems. *International Journal of Foundations of Computer Science*, 3(4):443–478, 1992.

28. Georg Struth and Krzysztof Ziemiański. Presheaf automata. *CoRR*, abs/2409.04612, 2024.

29. Rüdiger Valk. On the computational power of extended Petri nets. In Józef Winkowski, editor, *MFCS*, volume 64 of *Lecture Notes in Computer Science*, pages 526–535. Springer, 1978.

30. Rob J. van Glabbeek. The individual and collective token interpretations of Petri nets. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 323–337. Springer, 2005.

31. Rob J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science*, 356(3):265–290, 2006. See also [32].

32. Rob J. van Glabbeek. Erratum to "On the expressiveness of higher dimensional automata". *Theoretical Computer Science*, 368(1-2):168–194, 2006.

Appendix: proofs

Proof (of Lem. 4). Let $[\![N]\!]_1 = (V_N, E_N)$. We can define a bijection $f : V_N \to [\![N]\!]_0$ between V_N and 0-cells of $[\![N]\!]$ as $f(m) = (m, \emptyset)$ for all $m \in V_N$. Then, define $g : E_N \to [\![N]\!]_1$ such that $g((m_0, t, m_1)) = x = (m, t)$ if $\delta^0(x) = (m_0, \emptyset)$ and $\delta^1(x) = (m_1, \emptyset)$.

Thus, there exists $e = (m_0, t, m_1) \in E_N$ if and only if t can fire in m_0 *i.e.*, ${}^\bullet t \leq m_0$, and $m_1 = m_0 - {}^\bullet t + t^\bullet$ if and only if, by construction of $[\![N]\!]$, there exists $x \in [\![N]\!]_1$ and $m = m_0 - {}^\bullet t = m_1 - t^\bullet$ such that $x = (m, t)$. Thus $e = (m_0, t, m_1) \in E_N$ if and only if $g(e) = (f(m_0), (m, t), f(m_1))$. □

Proof (of Prop. 7). Note first that since N is bounded and T finite there exists a constant $k \in \mathbb{N}$ such that for every marking m, $m(p) \leq k$ for all $p \in S$. Thus the number of different reachable markings of $[\![N]\!]_1$ is finite. Besides, they correspond to the number of 0-cells of $\mathsf{ess}([\![N]\!])$ by Lem. 4. Now, by Lem. 5, proving that the number of edges of the reachable part of $[\![N]\!]_{\mathrm{CS}}$ is finite proves the proposition. Indeed, each edge (m, U, m') corresponds to a $|U|$-cell of $[\![N]\!]$, $U \in \square(T)$.

Since each $t \in T$ is constrained, *i.e.*, not preset-free, each t fires in some bounded marking m only if ${}^\bullet t \leq m$. Thus, for each reachable marking m there are finitely many $U \in \square(T)$ such that ${}^\bullet U \leq m$, and each such U leads to a unique marking $m' = m - {}^\bullet U + U^\bullet$. Note that these U may contain autoconcurrency but are finite since m is bounded. In addition, if (m, U, m') is an edge of $[\![N]\!]_{\mathrm{CS}}$, then there are $|U|!$ $|U|$-cells: $\{(m - {}^\bullet U' = m' - U'^\bullet, U') \mid U'$ a permutation of $U\}$ in $[\![N]\!]$ for which $(m, \emptyset)$ is a lower face and $(m', \emptyset)$ an upper face but only one edge $((m, \emptyset), U, (m', \emptyset))$ in $\mathsf{flat}([\![N]\!])$ by construction. Finally, since the reachable part of $[\![N]\!]_{\mathrm{CS}}$ is finite and contains finitely many edges, $\mathsf{ess}([\![N]\!])$ is finite by Lem. 5. □

Proof (of Lem. 12). The construction of $[\![N]\!]$ from $N = (S, T, F, I)$ starts by building the precubical set $X = [\![(S, T, F)]\!]$ and then take a subset $X' = \{(m, (t_1, ..., t_n)) \in \mathbb{N}^S \times \square \mid \forall i = 1, ..., n : \forall s \in {}^\circ t_i : m(s) = 0\}$. Thus we only need to show (4). That is if there exist $x_0, x_2 \in X'$ such that $\delta_{C,D}(\delta_{A,B}(x_0)) = x_2$ then $\delta_{A \cup C, B \cup D}(x_0) = x_2$. Let x_0, x_2 be such cells and let $x_1 \in X'$ such that $\delta_{A,B}(x_0) = x_1$ and $\delta_{C,D}(x_1) = x_2$. Assume $x_i = (m_i, U_i)$ and $U_i = (t_{i,1}, ..., t_{i,n_i})$. By construction of X' for all $t_{i,j} \in U_i$ and for all $s \in {}^\circ t_{i,j}$, $m_i(s) = 0$. As x_2 is already in X' it remains to show that $\delta_{A \cup C, B \cup D}(x_0) = x_2$, that is x_2 is reachable form x_0 by unstarting $A \cup C$ and terminating $B \cup D$. By definition of face maps $U_1 = U_0 \setminus (A \cup B)$ and $U_2 = U_1 \setminus (C \cup D)$. Thus $U_2 = U_0 \setminus (A \cup B \cup C \cup D)$. Moreover, by construction of X, $m_1 = m_0 + {}^\bullet A + {}^\bullet B$ and $m_2 = m_1 + C^\bullet + D^\bullet$. Hence $m_2 = m_0 + {}^\bullet A + {}^\bullet B + C^\bullet + D^\bullet$ and $\delta_{A \cup C, B \cup D}((m_0, U_0)) = (m_2, U_2)$. □