On Profiles and Footprints – Relational Semantics for Petri Nets

Matthias Weidlich[1] and Jan Martijn van der Werf* [2]

[1] Technion – Israel Institute of Technology, Haifa, Israel
weidlich@tx.technion.ac.il
[2] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
j.m.e.m.v.d.werf@tue.nl

Abstract. Petri net systems have been successfully applied for modelling business processes and analysing their behavioural properties. In this domain, analysis techniques that are grounded on behavioural relations defined between pairs of transitions emerged recently. However, different use cases motivated different definitions of behavioural relation sets. This paper focusses on two prominent examples, namely behavioural profiles and behavioural footprints. We show that both represent different ends of a spectrum of relation sets for Petri net systems, each inducing a different equivalence class. As such, we provide a generalisation of the known relation sets. We illustrate that different relation sets complement each other for general systems, but form an abstraction hierarchy for distinguished net classes. For these net classes, namely S-WF-systems and sound free-choice WF-systems, we also prove a close relation between the structure and the relational semantics. Finally, we discuss implications of our results for the field of business process modelling and analysis.

1 Introduction

Business process modelling emerged as a means to capture the operations of an organisation. A process model depicts the major activities conducted to achieve a certain goal along with their temporal dependencies [29]. Drivers for process modelling include, among others, the need to establish a shared understanding of the business processes, certification of operations, or process automation.

In practice, business process modelling is often conducted using domain-specific high-level languages, such as BPMN or EPCs, see [29]. For the analysis of process models, however, the Petri net formalism has been successfully employed for over a decade [1,6]. The simple yet powerful formalism is conceptual close to many of the industrial process languages and, in fact, inspired the definition of execution semantics for many of them. Also, the existing theory on the analysis

* supported by the PoSecCo project (project no. 257129), partially co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D (FP7)

of Petri nets proved to be valuable for answering many of the analysis questions for business process models, cf., [6].

Behavioural analysis of Petri net systems may be grounded on different types of semantics. For instance, the analysis of deadlock freedom of interacting business processes [30,4] suggests to consider the moment of choice, i.e., the state space of a system. Techniques from the field of process mining, in turn, are typically based on trace semantics [21,2,28]. Recently, sets of behavioural relations, which induce relational semantics for Petri nets, have been utilised for analysis, most prominently the behavioural profile [25] and the behavioural footprint [2]. These relations are defined between pairs of transitions and capture behavioural characteristics, or features in the data mining terminology [12], such as order and exclusiveness. The aforementioned notions of relational semantics are conceptual close. Both define relations that allow to represent the behavioural characteristics of a system as a matrix. Yet, they differ with respect to the captured characteristics since different utility considerations led to their definition. Even though the different relations sets proved to be useful in many use cases, their differences have not been thoroughly investigated so far. Insights into their relation and the induced equivalence classes, however, are needed to select a definition that is appropriate for a specific analysis setting.

In this paper, we address this need and make the following contributions. First, we show that the existing notions of profiles and footprints represent different ends of a spectrum of relation sets for Petri net systems. Based on this observation, we provide a generalization of the notion of a relation set. Second, we investigate the expressiveness of different relation sets in this spectrum. We illustrate that those complement each other for general systems. Third, we prove that relation sets form an abstraction hierarchy for classes of workflow (WF-)systems, i.e., S-WF-systems and sound free-choice WF-systems. For these systems, we also establish a link between the net structure and relational semantics. Finally, we elaborate on the implications of our investigations for the application of relational semantics in the field of business process modelling.

The remainder of this paper is structured as follows. The next section presents formal preliminaries. Sec. 3 generalises existing relation sets to obtain a spectrum of relational semantics. Sec. 4 elaborates on relation sets of distinguished net classes. Sec. 5 outlines implications of our work for the application of relation sets. Finally, we review related work in Sec. 6 and conclude in Sec. 7.

2 Preliminaries

Let S be a set. The powerset of S is denoted by $\mathcal{P}(S) = \{S' \mid S' \subseteq S\}$. We use $|S|$ for the number of elements in S. Two sets S and T are *disjoint* if $S \cap T = \emptyset$. A set of sets $U \subseteq \mathcal{P}(S)$ is a *partitioning* of S iff all sets in U are pairwise disjoint and $\bigcup_{X \in U} X = S$.

We denote the Cartesian product of two sets S and T by $S \times T$. A binary relation R from S to T is defined by $R \subseteq (S \times T)$. For $(x, y) \in R$, we also write $x\, R\, y$. For a relation $R \subseteq (S \times T)$, the inverse relation R^{-1} is defined as

$R^{-1} = \{(y,x) \in (T \times S) \mid x\,R\,y\}$. Let $R \subseteq (S \times S)$ be a binary relation over a set S. Relation R is reflexive if $x\,R\,x$ for all $x \in S$. It is irreflexive if $(x,x) \notin R$ for all $x \in S$. It is symmetric if $x\,R\,y$ implies $y\,R\,x$ for all $x, y \in S$, and asymmetric if relation R is not symmetric. The relation is antisymmetric if $x\,R\,y$ and $y\,R\,x$ imply $x = y$ for all $x, y \in S$.

A *bag* m over S is a function $m : S \to \mathbb{N}$, where $\mathbb{N} = \{0, 1, \ldots\}$ denotes the set of natural numbers. We denote e.g. the bag m with an element a occurring once, b occurring three times and c occurring twice by $m = [a, b^3, c^2]$. The set of all bags over S is denoted by $\mathbb{N}^S$. Sets can be seen as a special kind of bag where all elements occur only once; we interpret sets in this way whenever we use them in operations on bags. We use $+$ and $-$ for the sum and difference of two bags, and $=, <, >, \leq, \geq$ for the comparison of two bags, which are defined in a standard way.

A *sequence* over S of length $n \in \mathbb{N}$ is a function $\sigma : \{1, \ldots, n\} \to S$. If $n > 0$ and $\sigma(i) = a_i$ for $i \in \{1, \ldots, n\}$, we write $\sigma = \langle a_1, \ldots, a_n \rangle$. The length of a sequence is denoted by $|\sigma|$. The sequence of length 0 is called the *empty sequence*, and is denoted by ϵ. The set of all finite sequences over S is denoted by S^*. We write $a \in \sigma$ if a $1 \leq i \leq |\sigma|$ exists such that $\sigma(i) = a$. *Concatenation* of two sequences $\nu, \gamma \in S^*$, denoted by $\sigma = \nu;\gamma$, is a sequence defined by $\sigma : \{1, \ldots, |\nu|+|\gamma|\} \to S$, such that $\sigma(i) = \nu(i)$ for $1 \leq i \leq |\nu|$, and $\sigma(i) = \gamma(i-|\nu|)$ for $|\nu| + 1 \leq i \leq |\nu| + |\gamma|$.

Definition 1 (Petri Net). A *Petri net* is a tuple $N = (P, T, F)$ where P and T are finite disjoint sets of *places* and *transitions*, respectively, and $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation*. The set of all nodes $P \cup T$ is denoted by N.

For a node $n \in N$, we define its *preset* by ${}^\bullet n = \{m \mid (m,n) \in F\}$ and its *postset* by $n^\bullet = \{m \mid (n,m) \in F\}$. We lift the notion of presets (postsets) to sequences by ${}^\bullet\sigma = \bigcup_{n \in \sigma} {}^\bullet n$ ($\sigma^\bullet = \bigcup_{n \in \sigma} n^\bullet$) for $\sigma \in T^*$. A sequence $\pi \in N^*$ of length n is a *path* of N iff $(\pi(i), \pi(i+1)) \in F$ for all $1 \leq i < n$. The set of all paths of N from node $x \in N$ to node $y \in N$ is denoted by $\Pi(N)_{(x,y)}$. We assume all nets to be connected, i.e., for all Petri nets $N = (P, T, F)$ we assume $\Pi(N)_{(x,y)} \cup \Pi(N)_{(y,x)} \neq \emptyset$ for all nodes $x, y \in N$.

Definition 2 (System, Enabledness, Firing). Let $N = (P, T, F)$ be a Petri net. A *marking* of N is a bag over P, i.e., $m \in \mathbb{N}^P$. A Petri net $N = (P, T, F)$ with corresponding marking $m \in \mathbb{N}^P$ is called a *system*.

Let (N, m) be a system with $N = (P, T, F)$. A transition $t \in T$ is *enabled* in (N, m), denoted by $(N, m)[t\rangle$, if ${}^\bullet t \leq m$. An enabled transition can *fire*, resulting in a new marking $m' = m - {}^\bullet t + t^\bullet$, and denoted by $(N, m)[t\rangle(N, m')$.

Given a system (N, m_0) with $N = (P, T, F)$, we extend the firing rule to sequences in a standard way. A sequence $\sigma \in T^*$ is a *firing sequence* of (N, m_0) if markings $m_1, \ldots, m_n \in \mathbb{N}^P$ exist such that $(N, m_{i-1})[\sigma(i)\rangle(N, m_i)$ for all $1 \leq i \leq n$, and denoted by $(N, m_0)[\sigma\rangle(N, m_n)$.

The set of all traces from (N, m_0) is defined by $\mathcal{T}(N, m_0) = \{\sigma \in T^* \mid \exists m \in \mathbb{N}^P : (N, m_0)[\sigma\rangle(N, m)\}$, and the set of all *reachable markings* by $\mathcal{R}(N, m_0) =$

$\{m \mid \exists \sigma \in T^*, m \in \mathbb{N}^P : (N, m_0)[\sigma\rangle(N, m)\}$. Two systems (N, m_0) and (N', m'_0) are called *trace-equivalent* iff $\mathcal{T}(N, m_0) = \mathcal{T}(N', m'_0)$.

A transition $t \in T$ of a system (N, m_0) is *live*, iff for every marking $m \in \mathcal{R}(N, m_0)$ a reachable marking $m' \in \mathcal{R}(N, m)$ exists, such that $(N, m')[t\rangle$. If all transitions of (N, m_0) are live, the system is called *live*.

A place $p \in P$ of a system (N, m_0) is *k-bounded* for some $k \in \mathbb{N}$ iff $m(p) \leq k$ for every reachable marking $m \in \mathcal{R}(N, m_0)$. If all places of (N, m_0) are k-bounded, the system is called *k-bounded*. A system is called bounded if a $k \in \mathbb{N}$ exists such that N is k-bounded.

Let (N, m_0) be a system with $N = (P, T, F)$. The *concurrency relation* $\|_{co} \subseteq N \times N$ [13] contains all pairs of nodes (x, y) that are marked (in case of a place) or enabled (in case of a transition) concurrently in some reachable marking, i.e., $(x, y) \in \|_{co}$ iff a marking $m \in \mathcal{R}(N, m_0)$ exists such that $m \geq m_x + m_y$ with $m_j = [j]$ if $j \in P$ and $m_j = {}^\bullet j$ if $j \in T$. The concurrency relation is symmetric by definition.

On Petri nets, we define the following subclasses. A Petri net $N = (P, T, F)$ is called an *S-net* iff $|{}^\bullet t| \leq 1$ and $|t^\bullet| \leq 1$ for all transitions $t \in T$. Net N is called a *T-net* iff $|{}^\bullet p| \leq 1$ and $|p^\bullet| \leq 1$ for all places $p \in P$. It is called *free choice* iff ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$ implies ${}^\bullet t_1 = {}^\bullet t_2$ for all transitions $t_1, t_2 \in T$. Given a system (N, m), if N is an S-net (free-choice net), we call the system an S-system (FC-system)

A special subclass of Petri nets is the class of workflow nets. A Petri net $N = (P, T, F)$ is a *workflow net* (WF net) if two places $i, f \in P$ exist such that ${}^\bullet i = f^\bullet = \emptyset$ and all nodes are on a path from i to f, i.e., for each node $n \in N$, a path $\pi \in \Pi(N)_{(i,f)}$ exists such that $n \in \pi$. Place i is called the *initial place* of N, place f is called the *final place* of N. We define the short-circuited net of WF net N by $\bar{N} = (P, T \cup \{\bar{t}\}, F \cup \{(f, \bar{t}), (\bar{t}, i)\})$. WF net N is called *sound* if the system $(\bar{N}, [i])$ is live and bounded.

3 A Spectrum of Relational Semantics

This section outlines a spectrum of relational semantics, induced by a spectrum of sets of relations defined over pairs of transitions. We first give an overview of this spectrum in Sec. 3.1, before Sec. 3.2 presents the formal definition of parametrised relation sets. Sec. 3.3 elaborates on equivalences based on relation sets.

3.1 Overview

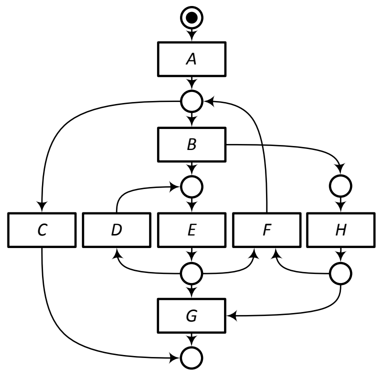
A first set of behavioural relations was presented as part of the α-mining algorithm [7,2]. It aims at the construction of a workflow net system from sequences of observed transition occurrences. To this end, it exploits direct successorship of transition occurrences, i.e., a *directly follows* relation. This relation comprises pairs of transitions that succeed each other. Using this relation, the α-algorithm defines three relations, $\#$ (or $+$ to harmonise notation), $\rightarrow$, and $\|$, over pairs of transitions. Membership of a transition pair for one of these relations is determined using the directly follows relation: two transitions may never follow each other (+), follow each other in one direction ($\rightarrow$), or in both directions ($\|$). As such, the relations (with relation $\rightarrow^{-1}$) partition the Cartesian product of the observed transitions. These relations are jointly referred to as a *footprint* [2]. Although proposed for sequences of observed transition occurrences, the relations may be derived for a net system based on all traces.

A similar, yet different set of behavioural relations was presented in [25], dubbed *behavioural profile*. The behavioural profile is grounded on the notion of *weak order*. This relation captures whether a transition is eventually succeeded by another transition. Again, three derived relations (+, $\rightarrow$, and $\|$ using the same notation) are constructed by investigating whether two transitions never occur together (+), are always ordered if they occur together ($\rightarrow$), or may occur in any order ($\|$). Together with the inverse of the order relation, one obtains a partitioning of the Cartesian product of transitions.

The footprint relations have been defined in the context of process mining. Here, the direct successorship of transitions according to the behavioural relations translates into a structural successorship during the construction of a net system. Behavioural profiles, in turn, have been motivated by the analysis of process model consistency. Models that shall be analysed for consistency typically show only a partial functional overlap, i.e., a certain share of transitions of one net system is without counterpart in the other system. Consistency measurement that is insensitive to such model extensions, therefore, is grounded on indirect behavioural dependencies as captured by the behavioural profile. With the same motivation, behavioural profiles have been applied for change propagation [27], process model abstraction [22], and the derivation of reusable modelling blocks [23].

Besides their differences, the footprint and the profile are conceptually close. Both adopt a binary base relation for transitions that requires the existence of a firing sequence that contains the respective transitions. The transitions are either required to succeed each other with no or an arbitrary number of transitions occurring in between. The difference, therefore, lies in the *look ahead* assumed to build the base relation. For illustration, consider the net system depicted in Fig. 1(a). Matrix $M1$ in Fig. 1(b) depicts the relational semantics induced by the footprint. It holds $D + F$, i.e., both transitions never succeed each other directly. In the footprint, relation + captures transitions that never succeed each other, whereas $\|$ captures concurrent enabling (e.g., $E \| F$) and control flow cycles of length one or two (e.g., $D \| E$). Matrix MF, in turn, shows the profile. Here, it holds $D \| F$ since the transitions may appear in either order. In the profile, relation + captures transitions that never occur together in any trace (e.g., $C + G$). Relation $\|$ captures concurrent enabling and control flow cycles of *any* length.

Both relation sets span a spectrum of relational semantics, which is obtained by step-wise increasing the assumed look ahead when constructing the base relation. We exemplify this spectrum by matrix $M2$. Here, the relations are derived from a *2-successor relation* that holds if two transitions follow each other



(a) An example net system.

(M1) Footprint 1-Successor

	A	B	C	D	E	F	G	H
A	+	→	→	+	+	+	+	+
B	←	+	+	+	→	←	+	→
C	←	+	+	+	+	←	+	+
D	+	+	+	+	∥	+	+	∥
E	+	←	+	∥	+	→	→	∥
F	+	→	→	+	←	+	+	←
G	+	+	+	+	←	+	+	←
H	+	←	+	∥	∥	→	→	+

(M2) 2-Successor

	A	B	C	D	E	F	G	H
A	+	→	→	+	→	+	+	→
B	←	+	+	→	∥	←	+	→
C	←	+	+	+	←	←	+	←
D	+	←	+	∥	∥	→	→	∥
E	←	∥	→	∥	∥	∥	→	∥
F	+	→	→	←	∥	+	+	∥
G	+	+	+	←	←	+	+	←
H	←	←	→	∥	∥	∥	→	+

(MF) Behavioural Profile Far-Successor

	A	B	C	D	E	F	G	H
A	+	→	→	→	→	→	→	→
B	←	∥	→	∥	∥	∥	→	∥
C	←	←	+	←	←	←	+	←
D	←	∥	→	∥	∥	∥	→	∥
E	←	∥	→	∥	∥	∥	→	∥
F	←	∥	→	∥	∥	∥	→	∥
G	←	←	+	←	←	←	+	←
H	←	∥	→	∥	∥	∥	→	∥

(b) Three different relational semantics.

Fig. 1. Overview of different relational semantics.

directly or with just a single transition occurring in between. In this matrix, we observe $D \rightarrow F$. There exists a trace in which D is succeeded by E and F. However, F is succeeded by D only once at least two transitions, B and E, have occurred.

3.2 Parametrised Relation Sets

In order to obtain a formalisation of the spectrum of relation sets, we first parametrise the base relation. The up-to-k successor relation holds between two transitions, if there exists a trace in which both transitions occur with at most $k-1$ transitions in between.

Definition 3 (k-Successor, up-to-k-Successor, minimal k-Successor). Let (N, m_0) be a system, let $T' \subseteq T$ be a set of transitions, and let $k \in \mathbb{N}$. The *k-successor relation* $\rhd_k \subseteq T' \times T'$ is defined by:

$$x \rhd_k y \Leftrightarrow \exists \sigma \in \mathcal{T}(N, m_0), 1 \leq i \leq |\sigma| : \sigma(i) = x \wedge \sigma(i+k) = y$$

The *up-to-k-successor relation* $>_k \subseteq T' \times T'$ is defined by:

$$x >_k y \Leftrightarrow \exists 1 \leq l \leq k : x \rhd_l y$$

The *minimal k-successor relation* $\trianglerighteq_k \subseteq T' \times T'$ is defined by:

$$x \trianglerighteq_k y \Leftrightarrow x \triangleright_k y \wedge (x, y) \notin >_{k-1}$$

Directly from the definition of the concurrency relation, if two transitions occur in the concurrency relation, then they are direct successors.

Proposition 4. Let (N, m_0) be a system, and let $x, y \in T$ be two transitions. If $(x, y) \in \|_{co}$ then $x >_1 y$.

Using the parametrised successor relation, we obtain parametrised relation sets.

Definition 5 (k-Relation set). Let $S = (N, m_0)$ be a system, let $T' \subseteq T$ be a set of transitions, and let $k \in \mathbb{N}$. Given a pair of transitions $(x, y) \in T' \times T'$, we define the k-exclusiveness relation $+_k \subseteq T' \times T'$, the k-order relation $\rightarrow_k \subseteq T' \times T'$ and the k-disorder relation $\|_k \subseteq T' \times T'$ by:

- $x +_k y$, iff $(x, y) \notin >_k$ and $(y, x) \notin >_k$;
- $x \rightarrow_k y$, iff $(x, y) \in >_k$ and $(y, x) \notin >_k$;
- $x \|_k y$, iff $(x, y) \in >_k$ and $(y, x) \in >_k$.

The k-relation set of S over T' is defined as a 3-tuple $\mathcal{S}_k^{T'}(N, m_0) = \{+_k, \rightarrow_k, \|_k\}$. If $T' = T$, we omit the superscript. We overload set comparison operators on relation sets by pairwise comparing the elements of the relation sets.

According to this definition, the footprint of a net system corresponds to its 1-relation set. We observe that the properties proved for footprints, see [2], hold also true for parametrised relation sets.

Property 6. Let (N, m_0) be a system, let $k \in \mathbb{N}$ and let $T' \subseteq T$ be a subset of transitions. Then for $\mathcal{S}_k^{T'}(N, m_0) = \{+_k, \rightarrow_k, \|_k\}$, it holds

(1) relation $\rightarrow_k$ is antisymmetric and irreflexive;
(2) relations $+_k$ and $\|_k$ are symmetric;
(3) $+_k$, $\rightarrow_k$ and $\|_k$ are pairwise disjoint; and
(4) $+_k$, $\rightarrow_k$, $\rightarrow_k^{-1}$ and $\|_k$ is a partitioning of $T' \times T'$.

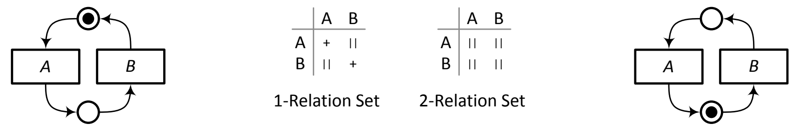
All properties follow directly from the definition of the relations based on the up-to-k-successor relation.

The parametrisation of relation sets induces an unbounded number of relational semantics for net systems. However, we observe that once a certain bound is reached, relation sets for higher parameters are all equal. We characterise this successor bound as follows.

Definition 7 (Successor bound). For a net system (N, m_0), the *successor bound* $b \in \mathbb{N}$ is the smallest number satisfying

- $\mathcal{S}_b(N, m_0) = \mathcal{S}_k(N, m_0)$ for all $b \leq k$; and
- $\mathcal{S}_k(N, m_0) \subset \mathcal{S}_b(N, m_0)$ for all $k < b$.

Proposition 8. Given a net system (N, m_0) there exists a unique successor bound with $b \leq |T|^2$.



(a) System S (b) Relation sets of S and S' (c) System S'

Fig. 2. Systems S and S' are k-equivalent for any $k \in \mathbb{N}$, but not trace equivalent.

Proof. Follows from the inductive definition of $>_k$ and $\triangleright_k$, and Prop. 6(4). $\square$

The successor bound for the relation sets of a net system is related to the notion of a behavioural profile. Apparently, the k-relation set with k being the successor bound coincides with the profile of a net system.

3.3 Equivalence of Relation Sets

Parametrised relation sets induce a number of equivalences. We first consider two types of equivalences. Two system may either show equivalent k-relation sets, or they even agree on all relation sets down to a certain boundary.

Definition 9 (k-Equivalent, down-to-k-equivalent). Let $S = (N, m_0)$ and $S' = (N', m'_0)$ be two systems, and let $\mathcal{S}_k(N, m_0) = \{+_k, \rightarrow_k, \|_k\}$ and $\mathcal{S}_k(N', m'_0) = \{+'_k, \rightarrow'_k, \|'_k\}$ be their respective k-relation sets.

- Systems S and S' are *k-equivalent*, denoted by $S \equiv_k S'$, iff $+_k = +'_k$, $\rightarrow_k = \rightarrow'_k$ and $\|_k = \|'_k$.
- Systems S and S' are *down-to-k-equivalent*, denoted by $S \equiv_{\downarrow k} S'$, iff $S \equiv_l S'$ for all $l \in \mathbb{N}$ with $k \leq l$.

Proposition 10. Relations $\equiv_k$ and $\equiv_{\downarrow k}$ are equivalences.

Proof. Relations $\equiv_k$ and $\equiv_{\downarrow k}$ are reflexive. Transitivity and symmetry follow directly from the set equivalences. $\square$

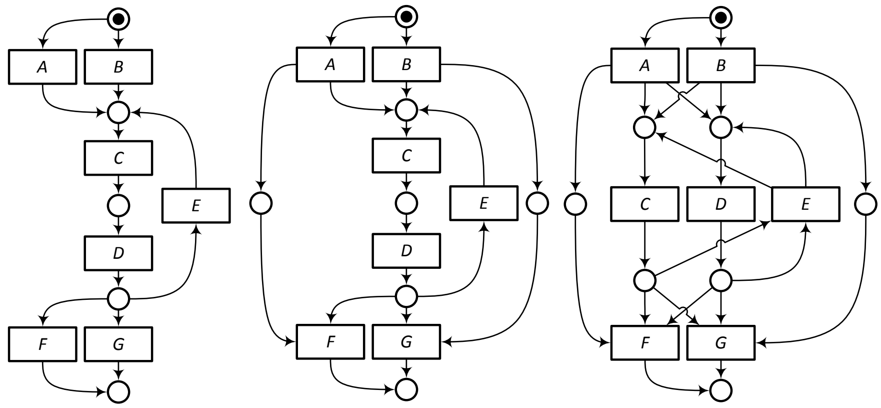
The relation sets are deduced from the up-to-k-successor relation, which formulates statements on the existence of a trace. As a consequence, net systems that show equal sets of traces show equal relation sets for all parameters.

Proposition 11. Let $S = (N, m_0)$ and $S' = (N', m'_0)$ be two systems that are trace equivalent. Then $S \equiv_{\downarrow 0} S'$.

Proof. Follows directly from the definition of $>_k$. $\square$

The opposite, however, does not hold. Consider Fig. 2. Systems S and S' are down-to-1-equivalent, i.e., the systems are k-equivalent for any $k \in \mathbb{N}$. However, as $\langle A, B, A \rangle$ is a firing sequence of S, but not of S', they are not trace equivalent.

Turning the focus on the relation between equivalence of relation sets for different parameters, we observe the following: Since relation sets beyond the successor bound do not change, equivalence for one parameter above this bound implies equivalence for all parameters above the bound.



(a) System S_1 (b) System S_2 (c) System S_3

Fig. 3. Equivalences based on different relation sets are incomparable; systems S_1 and S_2 are 1-equivalent, but not 7-equivalent; whereas S_2 and S_3 are 7-equivalent, but not 1-equivalent.

Theorem 12. *Let S and S' be two systems. Let b be the successor bound of S, and let b' be the successor bound of S'. Let $\bar{b} = \max\{b, b'\}$. If $S \equiv_{\bar{b}} S'$ then $S \equiv_{\downarrow \bar{b}} S'$.*

Proof. The proof follows directly from the definition of $\equiv_{\downarrow k}$ and Prop. 8. □

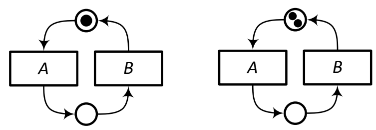
In general, equivalences based on different relation sets are incomparable. Consider for example the systems in Fig. 3. Systems S_1 and S_2 are 1-equivalent but not 7-equivalent. Likewise, systems S_2 and S_3 are 7-equivalent, but not 1-equivalent.

The example systems given in Fig. 2 illustrated already that different initial markings of a system may not be distinguished by relation sets. This is due to the fact that relation sets capture only the dependencies between transitions in terms of their minimal distance in any trace. However, they do not provide any notion of a *start* of a trace. To countervail this effect, we present two more equivalences. Those extend the given equivalences with the requirement of equal sets of initially enabled transitions.

Definition 13 (Start-k-equivalent, start-down-to-k-equivalent). Let $S = (N, m_0)$ and $S' = (N', m_0')$ be two systems. Let $T_0 = \{t \in T \mid (N, m_0)[t\rangle\}$ and $T_0' = \{t \in T' \mid (N', m_0')[t\rangle\}$ be the transitions enabled in the initial markings of both systems.

- Systems S and S' are *start-k-equivalent*, denoted by $S \equiv_k^s S'$ if $T_0 = T_0'$ and $S \equiv_k S'$.
- Systems S and S' are *start-down-to-k-equivalent*, denoted by $S \equiv_{\downarrow k}^s S'$, iff $S \equiv_l^s S'$ for all $l \in \mathbb{N}$ with $k \leq l$.

Proposition 14. Relations $\equiv_k^s$ and $\equiv_{\downarrow k}^s$ are equivalences.



(a) System S (b) System S'

Fig. 4. Systems S and S' are start-k-equivalent for any $k \in \mathbb{N}$, but not trace equivalent.

Proof. Relations $\equiv^s_k$ and $\equiv^s_{\downarrow k}$ are reflexive. Transitivity and symmetry follow directly from the set equivalences. □

As for the initially presented equivalences, the successor bound allows to draw conclusions on equivalence of relation sets for different parameters.

Proposition 15. Let $S = (N, m_0)$ and $S' = (N', m'_0)$ be two systems that are start-down-to-k-equivalent. Then, $S \equiv_{\downarrow k} S'$.

Proof. Follows from the definition of $\equiv^s_{\downarrow k}$. □

Also for these equivalences, however, the systems shown in Fig. 3 illustrate that this result does not hold in the general case.

Finally, we observe that, again, trace equivalence implies the equality of relation sets for all parameters and equality of sets of initially enabled transitions.

Proposition 16. Let $S = (N, m_0)$ and $S' = (N', m'_0)$ be two systems that are trace equivalent. Then $S \equiv^s_{\downarrow 0} S'$.

Proof. Follows from Prop. 11 and the definition of $\equiv^s_{\downarrow k}$. □

In contrast to k-equivalence, start-down-to-k-equivalence distinguishes the systems given in Fig. 2. However, even start-down-to-k-equivalence does not imply trace equivalence for general net systems, as illustrated by the systems in Fig. 4. Both systems are start-k-equivalent for any $k \in \mathbb{N}$, but $\langle A, A \rangle$ is a firing sequence of S, but not of S'. Thus, they are not trace equivalent.

4 Relation Sets for Distinguished Net Classes

This section investigates relation sets for distinguished classes of net systems, namely S-WF-systems and free-choice WF-systems. S-WF-systems provide a rather simple class of net systems, since they do not exhibit concurrency. As such, the structure of an S-WF-net is equivalent to its reachability graph. Further, the results for free-choice WF-systems imply those for S-WF-systems. Despite their simplicity and containment in the class of free-choice WF-systems, we first consider S-WF-systems to illustrate the investigated aspects. That is, first, the derivation of relation sets from the structure of the net system is shown. Second, we elaborate on the abstraction of a k-relation set, which yields a $k + 1$-relation set. Finally, we investigate the equivalence class induced by relation sets for the respective net systems.

4.1 S-WF-Systems

Derivation. In an S-WF-system, i.e., workflow systems that are also an S-net, we observe a close relation between the length of a directed path between two transitions and the fact whether they are k-successors. The reason for this close relation is the absence of concurrency in S-WF-systems.

Property 17. For a S-WF-system it holds $\|_{co}= \emptyset$.

This property which directly follows from the structure of S-WF-systems, allows for the following structural characterisation of k-successorship.

Lemma 18. *Let $S = (N, m_i)$ be an S-WF-system. Then, $x \rhd_k y$ iff there is a path $\pi \in \Pi(N)_{(x,y)}$ with $k = |\pi| - 1$.*

Proof. Marking m_i marks one place, so do all markings $m \in \mathcal{R}(N, m_i)$.
($\Rightarrow$) Let $x \rhd_k y$. Then, there exists a trace $\sigma \in \mathcal{T}(N, m_i)$ containing x and y, and $k - 1$ transitions between them. Since $\|_{co}= \emptyset$, the $k - 1$ transitions form a directed path.
($\Leftarrow$) Let there be a directed path comprising $k - 1$ transitions between x and y. Since every transition is on a path from the initial to the final place, all transitions have exactly one place in the preset and one place in the postset. Each reachable marking marks one place, hence, the $k - 1$ transitions may be fired in the respective order in any marking enabling x, so that $x \rhd_k y$. □

From the above, it follows that the derivation of a k-relation set requires only knowledge on the length of shortest directed paths between all transitions. Those may be determined in low polynomial time to the size of the system.

Theorem 19. *For any S-WF-system holds, any k-relation set is computed from its graph distance matrix.*

Proof. Follows directly from Lm. 18. □

Corollary 20. *Let (N, m_0) be a S-WF-system, and let $k \in \mathbb{N}$. The k-relation set can be calculated in $O(|N|^3)$.*

Proof. Follows from Thm. 19 and the fact that the shortest directed paths between all nodes of a directed graph with N nodes are determined in $O(|N|^3)$ time [24]. □

Abstraction. We introduce a notion of abstraction to describe the interplay between different relation sets of a single net system. Abstraction aims at deriving the $(k + 1)$-relation set from a k-relation set using the system structure. To this end, it extends the underlying successor relation. Since S-WF-systems do not show concurrency, abstraction can be done in a sequential way as follows.

Definition 21 (Sequential abstraction). Let (N, m_0) be a system with $N = (P, T, F)$, $>_k \subseteq T \times T$ an up-to-k successor relation. Then, the *sequential abstraction* of $>_k$, denoted by $\alpha^S(>_k) \subseteq T \times T$, is defined by $(x, z) \in \alpha^S(>_k)$, iff $x >_k z$ or a transition $y \in T$ exists with $x >_k y$ and $z \in p^\bullet$ for some $p \in y^\bullet$.

To show that abstraction of a k-relation set indeed yields the $(k+1)$-relation set, we first prove an auxiliary result. It states that structural precedence hints at the enabling of a transition in a certain marking.

Proposition 22. Let (N, m_0) be an S-WF-system with $N = (P, T, F)$. Let $\sigma \in T^*$ and $m \in \mathbb{N}^P$ such that $(N, m_0)[\sigma\rangle(N, m)$. Then $t \in p^\bullet$ for some $p \in \sigma(|\sigma|)^\bullet$ iff $(N, m)[t\rangle$ for any $t \in T$.

Proof. Follows from the structure of S-nets and the boundedness theorem [8]. □

Using this result, we can prove that abstraction indeed allows for generalising relation sets of S-WF-systems.

Proposition 23. Let (N, m_0) be an S-WF-system, and let $k \in \mathbb{N}$ such that $k > 0$. Then $\alpha^S(>_k) = >_{k+1}$.

Proof. Define $N = (P, T, F)$. It suffices to consider the case of $\triangleright_k$, as it implies the result for $>_k$. Let $x, z \in T$ such that $x \triangleright_k z$. Then, a trace $\sigma \in \mathcal{T}(N, m_0)$ and marking $m \in \mathbb{N}^P$ exist with $(N, m_0)[\sigma\rangle(N, m)$ and $\sigma(|\sigma| - k) = x$ and $\sigma(|\sigma|) = z$.
$(\Rightarrow)$ Consider a transition $y \in p^\bullet$ for some $p \in z^\bullet$. By Prop. 22, y may be fired in m, which yields $x \triangleright_{k+1} y$.
$(\Leftarrow)$ Assume σ is extended by firing a transition y in m. Then, $y \in p^\bullet$ for some $p \in z^\bullet$ by Prop. 22. □

Equivalence. Turning the focus on the equivalence classes induced by relation set for S-WF-systems, we observe that those form a hierarchy. A smaller parameter for the relation set yields a stricter equivalence.

Theorem 24. *Let S and S' be two S-WF-systems. If $S \equiv^s_1 S'$, then $S \equiv^s_k S'$, for any $k \in \mathbb{N}$ with $k > 0$.*

Proof. We prove the statement by showing that if $S \equiv^s_l S'$ for all $l \leq k$, then $S \equiv^s_{k+1} S'$. Let $(x, y) \in \alpha^S(>_k)$. Then either $(x, y) \in >_k$ or a $z \in T$ and $l < k$ exists with $(x, z) \in >_l$ and $z >_1 y$. Since $S \equiv^s_l S'$ and $S \equiv^s_1 S'$, both $(x, y) \in >'_l$ and $z >'_1 y$. Hence, $(x, y) \in \alpha^S(>'_k)$. By Prop. 23, we have $>_{k+1} = >'_{k+1}$. Hence, $S \equiv^s_{k+1} S'$. □

Finally, we show that 1-relation sets provide a complete characterisation of trace semantics for S-WF-systems. Hence, start-down-to-k-equivalence coincides with trace equivalence.

Theorem 25. *Let $S = (N, m_i)$ and $S' = (N', m'_i)$ be S-WF-systems. Then, S and S' are trace equivalent, iff $S \equiv^s_{\downarrow 1} S'$.*

Proof. Follows directly from Thm. 24, Prop. 22 and Prop. 16. □

4.2 Free-Choice WF-Systems

Free-choice Petri nets [8] is a well-studied subclass of Petri nets, for which many nice theoretical results and efficient algorithms exist. Many behavioural properties of free-choice nets are decidable based on the structure of the net, like well-formedness. In addition, free-choice nets are an important class for modelling business processes, since the essentials of common process description languages can be traced back to free-choice nets (exceptions include OR-joins and error handling) [17]. As an example, the BIT process library[3] contains 732 unique process models that all correspond to free-choice nets. In this section, we show that for free-choice nets, the up-to-k-successor can be decided on the structure of the net as well.

Derivation. For free-choice nets, we derive the up-to-k-successor using the minimal k-successor, i.e., the minimal k for which $x \rhd_k y$ holds. Based on the structure of free-choice nets, we introduce the minimal structural successor function (MSS). The MSS is a structural measure to calculate the number of steps needed to enable or mark a node y from a node x.

Definition 26 (Minimal structural successor). Let (N, m_0) be a system with $N = (P, T, F)$. The *minimal structural successor* $mss : N \times N \mapsto \mathcal{P}(N)$ assigns sets of nodes to pairs of nodes $x, y \in N$ as follows:

$$mss(x,y) = \begin{cases} \emptyset & \text{if } (x,y) \notin F^*, \\ \{x\} & \text{if } xF^*y,\ y \in x^\bullet, \\ \{x\} \cup \bigcup_{v \in x^\bullet, (v,y) \notin \|_{co}} mss(v,y) & \text{if } xF^*y,\ y \notin x^\bullet,\ x \in T, \\ \{x\} \cup mss(v,y) & \text{if } xF^*y,\ y \notin x^\bullet,\ x \in P,\ v \in x^\bullet, \\ & \ |mss(v,y)| = \min_{v \in x^\bullet} |mss(v,y)| \end{cases}$$

where F^* denotes the transitive closure of F.

For live and bounded free-choice systems, the minimal structural successor and the minimal k successor coincide, which means that we can compute the up-to-k successor in polynomial time. To prove this, we first show that a marking is reachable in which all necessary places needed to fire the transitions given by the minimal structural successor, i.e., places that are in the preset of these transitions, but not in their postset.

Proposition 27. Let (N, m_0) be a live and bounded system with $N = (P, T, F)$ free-choice. Let $x, y \in T$ such that $(x, y) \notin \|_{co}$ and $m_0(p) = 0$ for all places $p \in P \cap mss(x, y)$. Then a reachable marking $m \in \mathcal{R}(N, m_0)$ exists with $m \geq \sum_{p \in {}^\bullet U \setminus U^\bullet} [p]$ with $U = T \cap mss(x, y)$.

[3] http://www.zurich.ibm.com/csc/bit/downloads.html (last accessed March 26, 2012)

Proof. Since (N, m_0) is live, a marking $m \in \mathcal{R}(N, m_0)$ and firing sequence $\gamma \in (T \setminus U)^*$ exist such that $(N, m_0)[\gamma\rangle(N, m)$ and $(N, m)[x\rangle$, i.e., firing sequence γ enables transition x for the first time.

Define $Q(m) = \{p \in {}^\bullet\nu \setminus \nu^\bullet \mid m(p) > 0\}$, i.e., $Q(m)$ is the set of places marked in m needed to fire a transition of U, but not produced by any transition of U. Let $p \in {}^\bullet U \setminus (U^\bullet \cup {}^\bullet x) \setminus Q(m)$. As p is not in the postset of any transition in $mss(x, y)$, we have $(x, p) \in \|_{co}$. Since (N, m_0) is live, a firing sequence $\tau \in T^*$ and marking $m' \in \mathbb{N}^P$ exist such that $(N, m)[\tau\rangle(N, m')$, $m' \geq {}^\bullet x + [p]$ and $x \notin \tau$, since if x was needed to mark p, then $p \in mss(x, y)$. Suppose $Q(m) \cap {}^\bullet\tau \neq \emptyset$, i.e., some transition $u \in \tau$ consumes from some place $q \in Q(m)$. By the free-choice property, a transition $v \in mss(x, y)$ should then be enabled as well. However, x has not fired in $\gamma; \tau$, thus v cannot be enabled, and hence, u cannot be enabled as well. Thus, $Q(m) \cap {}^\bullet\tau = \emptyset$, and $Q(m') = Q(m) \cup \{q\}$. □

The above proposition shows the existence of a marking m in a free-choice system that generates sufficient tokens in order to fire the transitions of the MSS, as shown in the next proposition.

Proposition 28. Let (N, m_0) be a live and bounded system with $N = (P, T, F)$ is free-choice. Let $x, y \in T$ such that $(x, y) \notin \|_{co}$ and $m_0(p) = 0$ for all places $p \in P \cap mss(x, y)$. Then $x \rhd_k y$ with $k = |T \cap mss(x, y)|$.

Proof. We prove the statement by showing the existence of a firing sequence $\sigma \in \mathcal{T}(N, m_0)$ such that $\sigma(i) = x$ and $\sigma(i + k) = y$ for some $1 \leq i \leq |\sigma|$. We define relation $\sqsubset \subseteq T \times T$ by $a \sqsubset b$ if a $p \in mss(x, y) \cap P$ exists such that $\{(a, p), (p, b)\} \in F$, and $a \sqsubseteq b$ if either $a = b$ or a $c \in T \cap mss(x, y)$ exists such that $a \sqsubseteq c$ and $c \sqsubset b$. By definition, $\sqsubseteq$ is a partial order. Let $\nu \in T^*$ such that $\nu(i) \sqsubseteq \nu(j)$ for all $1 \leq i \leq j \leq |\nu|$ and $t \in \nu$ iff $t \in mss(x, y)$.

By Prop. 27, a firing sequence $\mu \in \mathcal{T}(N, m_0)$ and marking $m \in \mathbb{N}^P$ exist such that $(N, m_0)[\mu\rangle(N, m)$ and $m \geq \sum_{p \in {}^\bullet\nu \setminus \nu^\bullet}[p]$. Next, we show that ν is a firing sequence of (N, m). We prove this by induction on the length of ν.

Since $\nu(1) = x$, we have ${}^\bullet\nu(1) \subseteq \{p \in P \mid p \in {}^\bullet\nu \setminus \nu^\bullet\}$. Hence, ${}^\bullet\nu(1) \leq m$.

Now, suppose $\nu = \nu'; \nu''$ with $|\nu''| > 0$, and suppose a marking m' exists such that $(N, m)[\nu'\rangle(N, m')$. By construction of ν, $\nu'^\bullet \cap {}^\bullet u \neq \emptyset$. If ${}^\bullet u \subseteq \nu'^\bullet$, transition u is enabled in m'. Otherwise, a $p \in {}^\bullet u \setminus \nu'^\bullet$ exists. By the construction of ν, we have $m(q) > 0$.

Hence, firing sequence $\sigma = \mu; \nu$ has the desired property. □

Using this result, we establish the relation between the minimal k-successor relation and the minimal structural successor as follows.

Lemma 29. *Let (N, m_0) be a live and bounded system with $N = (P, T, F)$ free-choice. Let $x, y \in T$ such that $(x, y) \notin \|_{co}$ and $m_0(p) = 0$ for all places $p \in P \cap mss(x, y)$. Then $x \trianglerighteq_k y$ iff $|mss(x, y) \cap T| = k$.*

Proof. ($\Rightarrow$) Suppose $x \trianglerighteq_k y$ for some $k \in \mathbb{N}$. Then a firing sequence $\sigma \in \mathcal{T}(N, m_0)$ exists with an $1 \leq i \leq |\sigma|$ such that $\sigma(i) = x$ and $\sigma(i + k) = y$, and for all firing sequences $\tau \in \mathcal{T}(N, m_0)$ and $1 \leq i, j, \leq n$ such that $\tau(i) = x$ and $\tau(j) = y$, then $j - i \geq k$. As a consequence, the sequence $\langle \sigma(i), \ldots, \sigma(i+k) \rangle$ is cycle free, and each transition $\sigma(j)$ for $i \leq j \leq i+k$ is needed in order to enable y, since otherwise $x \leq_{k-1} y$. Hence, $(\sigma(j), y) \notin \|_{co}$ for all $i \leq j \leq i+k$. Thus, $\sigma(i) \in mss(x, y)$ for $i \leq j \leq i+k$, and $|mss(x, y) \cap T| \geq k$.

Suppose a $t \in mss(x, y) \cap T$ exists and no $i \leq j \leq i+k$ exists with $\sigma(j) = t$. Then $(t, y) \notin \|_{co}$ and firing y depends on firing transition t, i.e., a $j < i$ should exist with $\sigma(j) = t$, which is not possible since transition x needs to fire in order to enable t. Hence, such a transition does not exist and $|mss(x, y) \cap T| = k$.

($\Leftarrow$) Suppose $|mss(x, y) \cap T| = k$. To prove $x \trianglerighteq_k y$, we need to show $x \triangleright_k y$ and $(x, y) \notin >_{k-1}$. By Prop. 28, we have $x \triangleright_k y$.

Since $(x, y) \notin \|_{co}$, no marking exists in which both x and y are enabled. Suppose some place $p \in P \cap mss(x, y)$ is marked by some firing sequence not containing x, i.e., a firing sequence $\gamma \in \mathcal{T}(N, m_0)$ and marking $m \in \mathbb{N}^P$ exist with $(N : m_0 \xrightarrow{\gamma} m)$, $x \notin \gamma$ and $m(p) > 0$. Since (N, m_0) is live, a firing sequence $\sigma \in T^*$ and marking $m' \in \mathbb{N}^P$ exist such that $(N, m)[\sigma\rangle(N, m')$, $x \notin \sigma$ and $(N, m)[x\rangle$. Then $p \in \sigma$, since otherwise place p would be unbounded in (N, m_0). Thus, $m'(p) = 0$. Thus, all places $p \in P \cap mss(x, y)$ are empty when x fires.

Hence $(x, y) \notin >_{k-1}$. □

We conclude that the minimal structural successor suffices to compute the k-relation set of a live and bounded FC-system and, therefore, sound free-choice WF-systems.

Theorem 30. *Let (N, m_0) be a live and bounded FC-system. Then for any $k \in \mathbb{N}$, the k-relation set can be computed from its concurrency relation and its minimal structural successor.*

Proof. Follows from Prop. 4 and Lm. 29. □

Corollary 31. *Let (N, m_0) be a live and bounded FC-system, then for any $k \in \mathbb{N}$ with $k > 0$, the k-relation set can be calculated in $O(|N|^3)$ time.*

Proof. By Thm. 30, calculating the k-relation set for a live and bounded FC-systems can be done from its concurrency relation and the minimal successor length. The concurrency relation is computed in $O(|N|^3)$ time [13,11], the construction of the minimal structural successor also takes $O(|N|^3)$ time. □

Abstraction. As illustrated for S-systems, also for sound free-choice WF-systems there exists an abstraction operation for relation sets. Due to potential concurrency, the sequential abstraction introduced earlier cannot be applied, though. Therefore, we define a more generic abstraction operation based on the notion of the MSS.

Definition 32 (Abstraction). Let (N, m_0) be a system, $\|_{co} \subseteq T \times T$ its concurrency relation, and let $>_k \subseteq T \times T$ be an up-to-k successor relation. Then, the *abstraction* of $>_k$, denoted by $\alpha(>_k) \subseteq T \times T$, is defined by $(x, y) \in \alpha(>_k)$ iff $x >_k y$ or $k = |mss(x, y) \cap T| - 1$.

Abstraction, indeed, allows for deriving the $(k+1)$-relation set from the k-relation set for live and bounded free-choice systems and, therefore, for sound free-choice WF-systems. Sequential abstraction is a special case of abstraction for free-choice nets, as in S-systems, the MSS equals the shortest path between x and y.

Proposition 33. Let (N, m_0) be a live and bounded system with $N = (P, T, F)$ free-choice. Let $x, y \in T$ be two transitions such that $(x, y) \notin \|_{co}$, and $m_0(p) = 0$ for all places $p \in P \cap mss(x, y)$. Then $(x, y) \in \alpha(>_k)$ iff $(x, y) \in >_{k+1}$.

Proof. Define $l = |mss(x, y) \cap T|$.
($\Rightarrow$) Suppose $(x, y) \in \alpha(>_k)$. Then $(x, y) \in >_k$ or $k = 1 + |mss(x, y) \cap T|$. In the first case, also $(x, y) \in >_{k+1}$. Suppose not $(x, y) \in >_k$. Then $l = k - 1$ By Lm. 29, $x \trianglerighteq_l y$, i.e., $x \notin >_{l-1}$ and $x \triangleright_l y$. Hence, $x \triangleright_{k+1} y$.
($\Leftarrow$) Suppose $(x, y) \in >_{k+1}$. If $l < k + 1$, then $(x, y) \in >_k$. Otherwise, i.e., if $l = k + 1$, then, $k = |mss(x, y) \cap T| - 1$. Hence, $(x, y) \in \alpha(>_k)$. ☐

Equivalence. The abstraction for free-choice nets equals the $k + 1$ successor only if the places between the two nodes in consideration are initially empty. Therefore, we only consider free-choice WF-systems for equivalences. A sound WF-system is traced back to a live and bounded free-choice system and, initially, all places, except for the initial place, are empty.

Theorem 34. *Let S and S' be sound free-choice WF-systems. If $S \equiv^s_1 S'$, then $S \equiv^s_k S'$, for any $k \in \mathbb{N}$ with $k > 0$.*

Proof. Analogously to the proof of Thm. 24 using the abstraction of Def. 32 and Prop. 33. ☐

Finally, we consider the expressiveness of relation sets. For sound free-choice WF-systems, 1-relation sets provide a complete characterisation of trace semantics. Therefore, down-to-1-equivalence coincides with trace equivalence.

Theorem 35. *Let $S = (N, m_0)$ and $S' = (N', m'_0)$ be sound free-choice WF-systems. Then, S and S' are trace equivalent, iff $S \equiv^s_{\downarrow 1} S'$.*

Proof. Follows directly from Thm. 34, Prop. 28, and Prop. 16. ☐

5 Applications of Relational Semantics

Relation sets as proposed in this paper are a generalisation of existing notions of relational semantics. Earlier, we mentioned that those existing notions have been introduced for diverse applications within the field of business process modelling and analysis. We take up two applications and outline the benefits of using parametrised relation sets. Note that a formalisation of notions and measures for these applications is beyond the scope of this paper.

Process Model Similarity. Management of process model collections requires notions of process model similarity that are exploited for search and retrieval.

Recently, different relation sets have been used as a means for similarity measures, see [10,31,14]. The overlap of relations, e.g., determined by the Jaccard coefficient, is used to quantify behavioural similarity. However, those works rely on a single instantiation of relation sets, i.e., choose a certain parameter. On the one hand, this raises the question of how to choose the parameter since different utility considerations may be followed. On the other hand, more fine-granular measures can be obtained once more than one parameter, i.e., more than one relation set, is taken into account. Assume that two transitions are related by 1-order in one system $S1$, but not in another system $S2$. This negatively impacts the similarity score, independent of the fact whether the two transitions are related by 2-order or only by 20-order in $S2$. Taking the whole spectrum of parametrised relation sets (or a reasonable subset thereof) as the basis, therefore, enables to distinguish those cases. The difference between 1-order and 2-order is arguably less severe than the one between 1-order and 20-order, which can be reflected in the similarity score.

Conformance Analysis. Conformance analysis is an integral part of process mining. Given process logs that capture the observed execution sequences of a business process, conformance checking answers the question to which extent the observed behaviour is in line with the behaviour defined by an according business process model, i.e., a net system. Conformance analysis may be based on relation sets. Then, a relation set is constructed not only for a net system, but also for a single execution sequence or a complete process log, cf., [26,2]. Existing work leverages only a single instantiation of relation sets, which, again, raises the question of appropriateness of different relation sets. This holds in particular, since conformance analysis has to cope with alien events (events in a log that are not represented by any transition in a net system) and silent transitions (transitions of a net system that are not represented by any event). Parametrised relation sets provide a means to address these challenges: the parameter of a relation set may be stepwise increased until the relation for a pair of net transitions coincides with the relation of the corresponding log events. A low parameter hints at a less severe deviation compared to a high parameter.

We conclude that techniques defined for a single instantiation of relation sets may benefit from taking the whole spectrum of parametrised relation sets into account. Then, instead of basing the analysis on a fixed distance between transition occurrences, parametrised relation sets allow for additional insights into the differences between two behavioural models. It is important to see that this holds even if only sound free-choice WF-systems are considered. Although Sec. 4 showed that 1-relation sets provide a complete characterisation of trace semantics for this class, parametrised relation sets provide a means to quantify the severity of any deviation. If models that are no sound free-choice WF-systems are considered, the usage of parametrised relation sets is advantageous even in terms of expressiveness. As illustrated with the examples in Fig. 3, different relation sets are incomparable for general net systems. A detailed analysis on the expressiveness of parametrised relation sets for more general classes of net systems is left for future work, though.

6 Related Work

Our work relates to other types of relational semantics and their applications.

Work on process mining does not only rely on the footprint, i.e., the 1-relation set, but also considers other relations. A causal matrix has been used in genetic process mining [3]. It formulates dependencies for transitions using input and output functions that associate subsets of preceding or succeeding transitions to a single transition. Hence, they capture a share of the 1-relation set. To assess the quality of mined models, follows and precedes relations that associate the value 'never', 'sometimes', or 'always' to pairs of transitions have been proposed [21]. However, those relations neglect the distance between the occurrences of transitions. Closely related are also approaches to the declarative modelling of behaviour. Those allow for the restriction of possible behaviour by constraints, e.g., expressed in Linear Temporal Logic, see [5,18].

Relation sets capture complete trace semantics only for restricted system classes, but are a behavioural abstraction for general systems. Other approximations of trace semantics are causal footprints [9]. A causal footprint defines two relations, look-back links and look-ahead links. For a transition, those define a set of transitions of which one must have occurred before or after the transition. In contrast to relation sets, however, there is no unique causal footprint for a single system. Communication fingerprints [19] are another behavioural abstraction for net systems that focusses on boundaries and dependencies for the cardinalities of tokens consumed or produced at dedicated places. As such, this work can be seen to be orthogonal to relation sets.

Besides those mentioned in Sec. 5, applications of relation sets include the verification of hardware specifications, modelled as a labelled net system [20]. Here, relations that classify transitions of a net system as being sequential or parallel are used. Also, the management of business process variants has been addressed using an order matrix [15,16]. The latter is a specific instantiation of relation sets introduced in this paper.

7 Conclusions

In this paper, we took up existing definitions of behavioural relations and provided a generalisation that defines a spectrum of relational semantics for Petri net systems. The relational semantics induce equivalence classes for systems that are not comparable for general net systems. However, for sound free-choice WF-systems, we proved that the relation sets form a hierarchy. This allows to derive the $(k+1)$-relation set from a k-relation set with knowledge about the 1-relation set. Also, we showed that for this class of systems, relation sets are derived from the structure of net systems. Finally, the 1-relation set provides a complete characterisation of trace semantics for sound free-choice WF-systems.

These insights are valuable for applying relation sets for several reasons. First, if the goal is checking equivalence of sound free-choice WF-systems, it is sufficient to consider 1-relation sets. Second, once focus is on measuring behavioural differences, we outlined that even for sound free-choice WF-systems it is advantageous to consider different k-relation sets. Third, we illustrated that k-relation sets are incomparable for general net systems. Hence, for general net systems, the joint application of different relation sets allows for a closer approximation of trace semantics.

In future work, we aim at investigating the expressiveness of k-relation sets beyond the class of sound free-choice WF-systems. The example given in Fig. 3 illustrates that k-relation sets offer a means to distinguish behavioural differences that stem from non-free-choiceness. However, even a joint usage of different k-relation sets may not provide a complete characterisation of trace semantics for net systems. Thus, we aim at exploring for which more general classes of net systems the set of k-relation sets provides a complete trace characterisation. In addition, we aim at investigating k-relation sets for labelled net systems. Although we foresee that the relations may be directly lifted to labelled net systems, the influence on the expressiveness of k-relation sets needs to be clarified.

References

1. W.M.P. van der Aalst. The application of Petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21 – 66, 1998.
2. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.
3. W.M.P. van der Aalst, A.K. Alves de Medeiros, and A.J.M.M. Weijters. Genetic Process Mining. In *ICATPN 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 48 – 69. Springer-Verlag, Berlin, 2005.
4. W.M.P. van der Aalst, K.M. van Hee, P. Massuthe, N. Sidorova, and J.M.E.M. van der Werf. Compositional Service Trees. In *ICATPN 2009*, volume 5606 of *Lecture Notes in Computer Science*, pages 283 – 302. Springer-Verlag, Berlin, 2009.
5. W.M.P. van der Aalst, M. Pesic, and M.H. Schonenberg. Declarative workflows: Balancing between flexibility and support. *Computer Science - R&D*, 23(2):99 – 113, 2009.
6. W.M.P. van der Aalst and C. Stahl. *Modeling Business Processes: A Petri Net-Oriented Approach*. MIT Press, 2011.
7. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *Knowledge & Data Engineering*, 16(9):1128–1142, 2004.
8. J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1995.
9. B.F. van Dongen, R.M. Dijkman, and J. Mendling. Measuring Similarity between Business Process Models. In *CAiSE*, volume 5074 of *Lecture Notes in Computer Science*, pages 450 – 464. Springer-Verlag, Berlin, 2008.
10. R. Eshuis and P.W.P.J. Grefen. Structural Matching of BPEL Processes. In *ECOWS*, pages 171 – 180. IEEE Computer Society, 2007.
11. J. Esparza. A polynomial-time algorithm for checking consistency of free-choice signal transition graphs. *Fundamenta Informatica*, 62(2):197 – 220, 2004.
12. L. Huan and H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Springer-Verlag, Berlin, 1998.

13. A. Kovalyov and J. Esparza. A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs. In *Proc. of the International Workshop on Discrete Event Systems (WODES)*, pages 1 – 6. IEEE CS, 1996.
14. M. Kunze, M. Weidlich, and M. Weske. Behavioral Similarity - A Proper Metric. In *BPM 2011*, volume 6896 of *Lecture Notes in Computer Science*, pages 166 – 181. Springer-Verlag, Berlin, 2011.
15. C. Li, M. Reichert, and A. Wombacher. Discovering Reference Models by Mining Process Variants Using a Heuristic Approach. In *BPM 2009*, volume 5701 of *Lecture Notes in Computer Science*, pages 344 – 362. Springer-Verlag, Berlin, 2009.
16. C. Li, M. Reichert, and A. Wombacher. Representing block-structured process models as order matrices: Basic concepts, formal properties, algorithms. Technical report, University of Twente, Enschede, The Netherlands, 2009.
17. Niels Lohmann, Eric Verbeek, and Remco M. Dijkman. Petri net transformations for business processes - a survey. *T. Petri Nets and Other Models of Concurrency*, 2:46–63, 2009.
18. M. Montali, M. Pesic, W.M.P. van der Aalst, F. Chesani, P. Mello, and F. Storari. Declarative specification and verification of service choreographiess. *TWEB*, 4(1), 2010.
19. O.A. Oanea, J. Sürmeli, and K. Wolf. Service discovery using communication fingerprints. In *ICSOC*, volume 6470 of *Lecture Notes in Computer Science*, pages 612 – 618. Springer-Verlag, Berlin, 2010.
20. L.Ya. Rosenblum and Yakovlev. A.V. Analysing semantics of concurrent hardware specifications. In *Int. Conf. on Parallel Processing (ICPP89)*, pages 211 – 218, 1989.
21. A. Rozinat and W.M.P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.*, 33(1):64 – 95, 2008.
22. S. Smirnov, M. Weidlich, and J. Mendling. Business process model abstraction based on synthesis from well-structured behavioral profiles. *International Journal of Cooperative Information Systems (IJCIS),*, 2012. To appear.
23. S. Smirnov, M. Weidlich, J. Mendling, and M. Weske. Action patterns in business process model repositories. *Computers in Industry (COMIND)*, 63(2):98–111, 2012.
24. S. Warshall. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11 – 12, 1962.
25. M. Weidlich, J. Mendling, and M. Weske. Efficient consistency measurement based on behavioral profiles of process models. *IEEE Trans. Software Eng.*, 37(3):410 – 429, 2011.
26. M. Weidlich, A. Polyvyanyy, N. Desai, J. Mendling, and M. Weske. Process compliance analysis based on behavioural profiles. *Inf. Syst.*, 36(7):1009 – 1025, 2011.
27. M. Weidlich, M. Weske, and J. Mendling. Change propagation in process models using behavioural profiles. In *IEEE SCC*, pages 33 – 40. IEEE CS, 2009.
28. J.M.E.M. van der Werf, B.F. van Dongen, C.A.J. Hurkens, and A. Serebrenik. Process Discovery Using Integer Linear Programming. *Fundamenta Informatica*, 94(3 – 4):387 – 412, 2009.
29. M. Weske. *Business Process Management: Concepts, Languages, Architectures.* Springer-Verlag, Berlin, 2007.
30. K. Wolf. Does My Service Have Partners? *Transactions on Petri Nets and Other Models of Concurrency II*, 5460:152 – 171, 2009.
31. H. Zha, J. Wang, L. Wen, C. Wang, and J. Sun. A workflow net similarity measure based on transition adjacency relations. *Computers in Industry*, 61(5):463 – 471, 2010.