Exploring Complexity: An Extended Study of Formal Properties for Process Model Complexity Measures

Patrizia Schalk[1], Adam Burke[2], and Robert Lorenz[1]

[1] University of Augsburg, Universitätsstraße 6a, 86159 Augsburg, Germany
`{patrizia.schalk,robert.lorenz}@uni-a.de`
[2] Queensland University of Technology, 2 George St, 4000 Brisbane City, Australia
`at.burke@qut.edu.au`

Abstract. A good process model is expected not only to reflect the behavior of the process, but also to be as easy to read and understand as possible. Because preferences vary across different applications, numerous measures provide ways to reflect the complexity of a model with a numeric score. However, this abundance of different complexity measures makes it difficult to select one for analysis. Furthermore, most complexity measures are defined for BPMN or EPC, but not for workflow nets.
This paper adapts existing complexity measures to the world of workflow nets. It then compares these measures with a set of properties originally defined for software complexity, as well as new extensions to it. We discuss the importance of the properties in theory by evaluating whether matured complexity measures should fulfill them or whether they are optional. We find that not all inspected properties are mandatory, but also demonstrate that the behavior of evolutionary process discovery algorithms is influenced by some of these properties. Our findings help analysts to choose the right complexity measure for their use-case.

1 Introduction

Process discovery concerns finding a model for a business process [1]. The goal is to automatically construct an understandable model that contains all relevant behavior, so stakeholders and process analysts can make business decisions. Especially large processes that contain many sub-processes tend to produce cluttered and complex models. This is one of the reasons why the Directly Follows Miner [2] is so popular in practice: Directly Follows Graphs have only one type of node and easy semantics for the arrows. However, this model type struggles with distinguishing concurrency from loops. BPMN, EPCs and workflow nets, on the other hand, feature easy ways to model concurrency but have several types of nodes that each have different semantics. Consequently, researchers have developed complexity measures for these model types to evaluate which combinations of nodes make models difficult to understand. These measures tend to count one type of complex structure in a model. Because there are many types of such structures, numerous complexity measures were proposed [3]. Formal properties
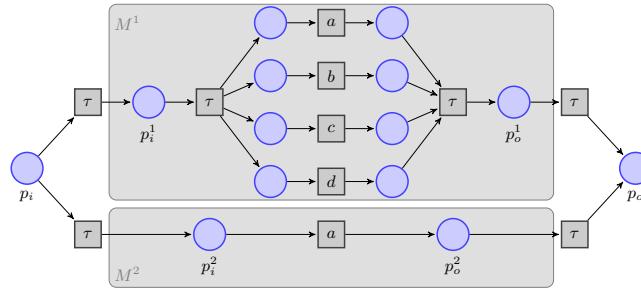
Fig. 1: A workflow net M with 14 places and 11 transitions.

for such measures help us understand their behavior. Creators of new measures can and should clearly state which properties the measure satisfies, and designers of discovery algorithms may prove desirable simplicity properties are always maintained. The same is true for software programs, so as a step towards this ideal setting, Weyuker [4] introduced nine formal properties that complexity measures for software should fulfill. We find six of these properties useful for process models as well, while three are less meaningful for this setting. We also draw on work from process mining on properties for complexity [5] and other quality dimensions [6], and propose new properties. Since process discovery algorithms benefit most from formal properties for complexity measures, we perform the analyses on workflow nets, like that in Figure 1, which have a strong theory and are therefore often returned by discovery algorithms. Our analyses hold for BPMN and EPCs as well, since workflow nets can be translated into these modeling languages [7]. The paper is structured as follows: Section 2 gives an overview of the analyses already performed on complexity measures, before Section 3 presents the basic definitions we need for our analyses. Section 4 then defines the properties that we use for the comparative analysis of the complexity measures defined in Section 5. Section 5 performs the detailed analyses. In Section 6, we show and discuss the implications for the properties, complexity measures and evolutionary discovery algorithms. Section 7 concludes the paper.

2   Related Work

There are many factors that make a process model difficult to understand. Mendling [3] identified 28 such factors and defined complexity measures for EPC models based on them. He demonstrated their relevance by showing that these complexity measures predict modeling errors in the SAP reference. Reijers et al. [8] found that some of these measures are also tied to the understandability of EPC models by conducting a case study with students of three universities. According to their study, especially the number of nodes, the density of the network, the average degree of connectors and the cross-connectivity between nodes influence how understandable a process model is. Lieben et al. [9] analyzed if existing complexity measures often agree in their score. They found that they

are not completely distinct, but essentially cover four dimensions of complexity, which they call TOKEN BEHAVIOR COMPLEXITY, NODE IO COMPLEXITY, PATH COMPLEXITY and DEGREE OF CONNECTEDNESS. They conclude that it is not necessary to use every complexity measure to adequately evaluate the complexity of a model. Instead, it is sufficient to use one complexity measure of each dimension, as well as two measures that do not fit to any dimension.

Until now, little is known about formal properties of complexity measures. For the quality dimensions fitness, precision and generalization, Syring et al. [10] use desirable properties defined by van der Aalst [6] to compare measures. Since they focus on the behavior and abstract from the representation of the model, they did not include properties for complexity measures. Weyuker [4], on the other hand, focuses on properties for software complexity measures. Because software and process models use similar control structures, these properties are a good starting point to analyze complexity measures for process models. Cardoso[5] therefore analyzed the control flow complexity with Weyuker's properties.

Whether a complexity measure fulfills formal properties is important for algorithms that find process models by optimizing over quality criteria. An example of such a discovery algorithm is the Evolutionary Tree Miner (ETM) [11]. This evolutionary algorithm continuously mutates a randomly generated set of process models until a model with good fitness, precision, generalization, and simplicity is found. The importance of each dimension can be set by weights, and the concrete measures can be chosen freely. In turn, the results of the ETM depend on the chosen quality measures and their properties. Because properties of the quality dimensions fitness, precision, and generalization are already analyzed, we focus on evaluating whether Weyuker's properties [4] are useful for complexity measures of process models.

3 Preliminaries

Let $\mathbb{N} := \{1, 2, \dots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Further, let $\mathbb{R}$ be the set of real numbers and $\mathbb{R}_0^+ := \{r \in \mathbb{R} \mid r \geq 0\}$. To define workflow nets, we first need a definition of Petri nets.

Definition 1 (Petri net). *A (simple)* Petri net *is a 3-tuple* $N = (P, T, F)$, *where P is a finite set of* places, *T is a finite set of* transitions, *$P \cap T = \emptyset$ and F is a* flow relation *with $F \subseteq (P \times T) \cup (T \times P)$. For any place $p \in P$ of N, we call $^\bullet p = \{t \in T \mid (t, p) \in F\}$ the* preset *of p and $p^\bullet = \{t \in T \mid (p, t) \in F\}$ the* postset *of p. We define the pre- and postset of transitions $t \in T$ accordingly.*

Workflow nets are frequently used for the modelling of business processes. In this paper, we focus on *labeled workflow nets*—a special subclass of Petri nets—for our analyses of complexity measures.

Definition 2 (Labeled workflow nets). *A labeled* workflow net *is a 6-tuple* $W = (P, T, F, \ell, p_i, p_o)$ *where (P, T, F) is a Petri net, $p_i, p_o \in P$ with $p_i \neq p_o$,*

  – *p_i is the only place in W for which $|^\bullet p_i| = 0$,*

- p_o *is the only place in W for which $|p_o^\bullet| = 0$,*
- *every node in $P \cup T$ lies on some path from p_i to p_o*

and, for a finite alphabet A with $\tau \notin A$, $\ell : T \to A \cup \{\tau\}$ is a labeling function, assigning a transition label to each transition. We call $t \in T$ with $\ell(t) = \tau$ a silent transition.

With $\mathcal{M}$, we denote the set of all possible workflow nets. Figure 1 and Figure 2 illustrate workflow nets with their places drawn as circles, and transitions drawn as squares. When the labeling of transitions doesn't matter, we leave these squares empty. Otherwise, we write the label of the transition in its center.

Fig. 2: The smallest possible workflow net, W_0.

Many complexity measures take connectors for choices or parallelism into account. Therefore, to make our analyses also valid for BPMN and EPC models, we define which structures in a workflow net we understand as a connector.

Definition 3 (Connectors in workflow nets). *Take an arbitrary workflow net $W = (P, T, F, \ell, p_i, p_o)$, a transition $t \in T$ and a place $p \in P$.*

- *If $|p^\bullet| > 1$, we call p an* xor-split.
- *If $|^\bullet p| > 1$, we call p an* xor-join.
- *If $|t^\bullet| > 1$, we call t an* and-split.
- *If $|^\bullet t| > 1$, we call t an* and-join.

Accordingly, we define

- *The set of xor-splits in W as $\mathcal{S}_{xor}^W := \{p \in P \mid |p^\bullet| > 1\}$,*
- *the set of xor-joins in W as $\mathcal{J}_{xor}^W := \{p \in P \mid |^\bullet p| > 1\}$,*
- *the set of and-splits in W as $\mathcal{S}_{and}^W := \{t \in T \mid |t^\bullet| > 1\}$,*
- *the set of and-joins in W as $\mathcal{J}_{and}^W := \{t \in T \mid |^\bullet t| > 1\}$.*

Note that these sets are not necessarily disjoint. The set of xor-connectors in W is $\mathcal{C}_{xor}^W := \mathcal{S}_{xor}^W \cup \mathcal{J}_{xor}^W$, the set of and-connectors in W is $\mathcal{C}_{and}^W := \mathcal{S}_{and}^W \cup \mathcal{J}_{and}^W$ and the set of all connectors is $\mathcal{C}^W := \mathcal{C}_{xor}^W \cup \mathcal{C}_{and}^W$.

In Definition 3, we don't define or-connectors for workflow nets. This is because there are multiple ways to model an or-connector in this model type. However, all of these ways use parallel splits and exclusive choices, so if a complexity measure punishes a connector representing an inclusive choice, it can punish those connectors instead. The complexity of a model may not only depend on the connector types but also on the labeling function, leading to the next definition.

Definition 4 (Relabeling of a Workflow net). *Let A and B be two finite alphabets. Let $W \in \mathcal{M}$ be a labeled workflow net $W = (P, T, F, \ell, p_i, p_o)$ with $\ell : T \to A \cup \{\tau\}$. We call a labeling $\ell' : T \to B \cup \{\tau\}$ a* uniform relabeling *of ℓ if for all transitions $t_1, t_2 \in T : \ell(t_1) = \ell(t_2) \leftrightarrow \ell'(t_1) = \ell'(t_2)$ and for all transitions $t \in T : \ell(t) = \tau \leftrightarrow \ell'(t) = \tau$. We denote the set of all uniform relabelings of ℓ by $\mathcal{R}_\ell$. For any $\ell' \in \mathcal{R}_\ell$, we call $W_{\ell'} = (P, T, F, \ell', p_i, p_o)$ a* relabeling *of W.*

For one of Weyuker's properties, we want to check if changing the order of activities impacts a complexity measure. Cardoso [5] implements this by allowing activities to change labels, and connectors to change type. We exploit the expressiveness of workflow nets and define a permutation as a net with the same transitions, but different control flow. Thus, we get closer to what Weyuker defined as permutations for software programs.

Definition 5 (Permutations of Workflow Nets). *Let W be a workflow net with $W = (P, T, F, \ell, p_i, p_o)$, P' be a set of places, $F' \subseteq (P' \times T) \cup (T \times P')$ be an arbitrary flow relation and $p_i', p_o' \in P'$ two places in P'. Using these notions, we call $Perm(W) := \{W' = (P', T, F', \ell, p_i', p_o') \mid W' \text{ is a workflow net}\}$ the set of* permutations *of W.*

We also want to investigate what happens if we combine two or more models. For the operations, we take inspiration from the creation of block-structured workflow nets [12] and use these operations on arbitrary nets. The operations defined in Definition 6 include sequential composition ($\to$), parallel composition ($\wedge$), exclusive choice ($\times$) and iteration ($\circlearrowleft$).

We have two reasons to focus on these operations: First, they are very well known and established due to the inductive miner [12],[1, p.222] relying on them, so many process analysts are already familiar with them. Second, the ETM heavily relies on these operations, which gives us the opportunity to directly test the impact of certain properties on a well-established discovery algorithm. The ETM does not operate directly on workflow nets, but on *process trees* [1], which can easily be converted to workflow nets with the operations of Definition 6.

Definition 6 (Operations on workflow nets). *Let $n \in \mathbb{N}$ with $n \geq 2$. For $j \in \{1, \dots, n\}$, let $M^j = (P^j, T^j, F^j, p_i^j, p_o^j)$ be workflow nets with disjoint sets of places and transitions ($P^1 \cap \dots \cap P^n = \emptyset$ and $T^1 \cap \dots \cap T^n = \emptyset$). We define*

- *The* sequential composition $\to (M^1, \dots, M^n) := (P_\to, T_\to, F_\to, p_i^1, p_i^n)$, *with*
  - $P_\to := \bigcup_{j=1}^n P^j$,
  - $T_\to := \bigcup_{j=1}^n T^j \cup \bigcup_{j=1}^{n-1} \{t_j^*\}$ *for new transitions $t_1^*, \dots, t_{n-1}^*$ and*
  - $F_\to := \bigcup_{j=1}^n F_j \cup \bigcup_{j=1}^{n-1} \{(p_o^j, t_j^*), (t_j^*, p_i^{j+1})\}$.
- *The* parallel composition $\wedge (M_1, \dots, M_n) := (P_\wedge, T_\wedge, F_\wedge, p_i^*, p_o^*)$, *where p_i^* and p_o^* are new places and*
  - $P_\wedge := \bigcup_{j=1}^n P^j \cup \{p_i^*, p_o^*\}$,
  - $T_\wedge := \bigcup_{j=1}^n T^j \cup \{t_i^*, t_o^*\}$ *for new transitions t_i^* and t_o^* and*
  - $F_\wedge := \bigcup_{j=1}^n F^j \cup \{(p_i^*, t_i^*), (t_o^*, p_o^*)\} \cup \bigcup_{j=1}^n \{(t_i^*, p_i^j), (p_o^j, t_o^*)\}$.

- *The* choice $\times (M_1, \ldots, M_n) := (P_\times, T_\times, F_\times, p_i^*, p_o^*)$, *where* p_i^* *and* p_o^* *are new places and*

  - $P_\times := \bigcup_{j=1}^n P^j \cup \{p_i^*, p_o^*\}$,
  - $T_\times := \bigcup_{j=1}^n (T^j \cup \{t_j^*, s_j^*\})$ *and*
  - $F_\times := \bigcup_{j=1}^n (F^j \cup \{(p_i^*, t_j^*), (t_j^*, p_i^j), (p_o^j, s_j^*), (s_j^*, p_o^*)\}$.

- *The* iteration $\circlearrowleft (M_1, \ldots, M_n) := (P_\circlearrowleft, T_\circlearrowleft, F_\circlearrowleft, , p_i^*, p_o^*)$, *where* p_i^* *and* p_o^* *are new places and*

  - $P_\circlearrowleft := \bigcup_{j=1}^n P^j \cup \{p_i^*, p_o^*, p^*, q^*\}$,
  - $T_\circlearrowleft := \bigcup_{j=1}^n (T^j \cup \{t_j^*, s_j^*\}) \cup \{t^*, s^*\}$ *and*
  - $F_\circlearrowleft := \{(p_i^*, t^*), (t^*, p^*), (q^*, s^*), (s^*, p_o^*)\} \cup \bigcup_{j=1}^n F_j$
    $\cup \bigcup_{j=1}^{n-1} \{(p^*, t_j^*), (t_j^*, p_i^j), (p_o^j, s_j^*), (s_j^*, q^*)\}$
    $\cup \{(q^*, s_n^*), (s_n^*, p_i^n), (p_o^n, t_n^*), (t_n^*, p^*)\}$.

Figure 3 shows how these operations combine workflow nets.

(a) The sequential composition $\rightarrow$.

(b) The parallel composition $\wedge$.

(c) The choice operator $\times$.

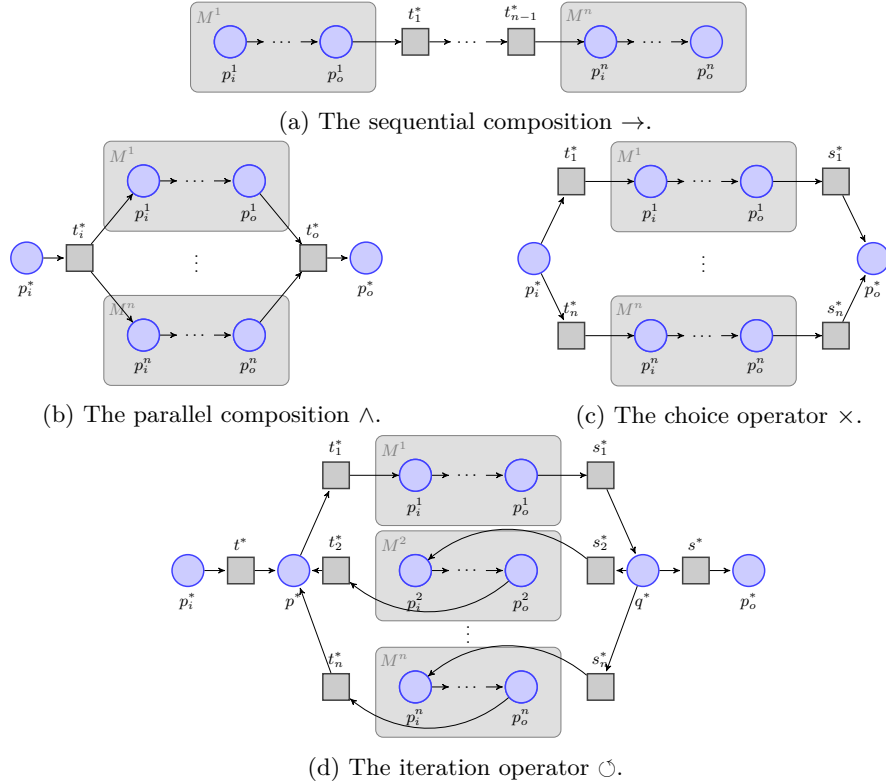(d) The iteration operator $\circlearrowleft$.

Fig. 3: A schematic overview of the four operations of Definition 6.

Finally, for this section, we define complexity measures. Existing complexity measures all are functions that take a model as input and return a *complexity score*—a real value that reflects how complex the net is.

Definition 7 (Complexity Measure). *Let $\mathcal{M}$ be the set of all possible workflow nets. A* complexity measure C *is a function* $C : \mathcal{M} \to \mathbb{R}$.

Complexity influences many properties of a process model, like its understandability, correctness, or the time needed to execute certain algorithms.

4 Properties of Complexity Measures

In the first part of this section, we define Weyuker's properties for workflow nets. Since this set of properties is not intended to be complete, we propose simple extensions that further deepen the understanding of the analyzed complexity measures in a second part. In both parts, we take C as a placeholder for any complexity measure.

4.1 Properties of Weyuker

The first property states that a complexity measure should be able to return more than a single score. This property is important independent of the use-case, since a complexity measure that assigns the same score to all input is not useful for any purpose regarding complexity.

Property 1 ($\neg$TRIV (W1)) *C is **not trivial**, i.e. it doesn't assign the same complexity score to all process models:*

$$\exists M_1, M_2 \in \mathcal{M} : C(M_1) \neq C(M_2)$$

A non-trivial measure can still be of little use if the amount of assigned scores is low. Property two therefore states only finitely many inputs should receive the same complexity score.

Property 2 (FIN (W2)) *There are only **finitely many** process models that receive the same complexity score from C:*

$$\forall c \in \mathbb{R} : |\{M \in \mathcal{M} \mid C(M) = c\}| < \infty$$

If a complexity measure assigns a unique score to each workflow net, FIN is fulfilled, but the measure is too specific: It would find a numerical representation of the input instead of rating its complexity. This is reflected in the third property.

Property 3 (COL (W3)) *C allows for **collisions**, i.e. different process models can get the same complexity score:*

$$\exists M_1, M_2 \in \mathcal{M} : M_1 \neq M_2 \wedge C(M_1) = C(M_2)$$

Weyuker states that not the output of a program should influence its complexity, but the details of the implementation [4]. Accordingly, the modeled process should not define the complexity of the model. Put differently, if we have two models with the same language, it should be possible for these models to get different complexity scores. For a workflow net $M \in \mathcal{M}$, we define its language $L(M)$ as done in [1].

Property 4 (IND$_L$ (W4)) *C is **independent** of the input model's language:*

$$\exists M_1, M_2 \in \mathcal{M} : L(M_1) = L(M_2) \wedge C(M_1) \neq C(M_2)$$

It seems sensible that a small part of a process model should not be more complex than the entire model. For example, the complexity of the model M^1 of Figure 1 should not be higher than the complexity of the entire model, M. Weyuker therefore proposes that complexity measures should be monotone in this sense. Van der Aalst [6] agrees with this, as his properties **RecPro3**, **PrecPro1** and **GenPro3** are closely related to MON.

Property 5 (MON (W5)) *C is **monotone**, i.e. the complexity score of a composed model cannot be lower than the complexity score of one of its parts: Let $\mathcal{O} := \{\rightarrow, \wedge, \times, \circlearrowleft\}$. For any $M_1, \ldots, M_n \in \mathcal{M}$ we have:*

$$\forall \oplus \in \mathcal{O} : C(\oplus(M_1, \ldots, M_n)) \geq C(M_1)$$

For the next property, suppose we have two models with the same complexity score. Weyuker argues that composing each of them with a third one does not necessarily lead to the same complexity score, since one part of a model can influence the complexity of another part. We can observe this with the nets in Figure 1: The fact that the activity a of M^2 is also present in M^1 increases the perceived complexity of M. If M^1 would not contain the activity label a, the net would be simpler. It is desirable that a complexity measure finds such dependencies.

Property 6 (COMP (W6)) *C is **composition sensitive**, i.e. there are process models of equal complexity according to C, whose complexity differ when composed with a third process model. Let $\mathcal{O} := \{\rightarrow, \wedge, \times, \circlearrowleft\}$.*

$$\forall \oplus \in \mathcal{O} : \exists M_1, M_2, M_3 \in \mathcal{M} : C(M_1) = C(M_2) \wedge C(M_1 \oplus M_3) \neq C(M_2 \oplus M_3)$$

One way of transforming a program is to reorder its instructions. Weyuker argues that such a reordering should impact the complexity. This is also true for process models: If we add more arcs to M in Figure 1, it looks more complex.

Property 7 (PERM (W7)) *C is sensitive for **permutations**, i.e. changing the start- and end-points of arcs in a workflow net can have an impact on the complexity score:*

$$\exists M \in \mathcal{M} : \exists M' \in Perm(M) : C(M) \neq C(M')$$

Weyuker states that renaming the variables of a program should not impact its complexity. Cardoso [5] translates the renaming of variables to the uniform renaming of activity names. We also use this translation for workflow nets.

Property 8 (ROB_ℓ (W8)) *C is **robust** against relabelings, i.e. uniformly changing the labeling does not affect complexity:*

$$\forall M = (P, T, F, \ell, p_i, p_o) \in \mathcal{M} : \forall \ell' \in \mathcal{R}_\ell : C(M) = C(M_{\ell'})$$

The final property concerns the combination of two process models. According to Weyuker, it should be possible that the complexity of a composed model is higher than the sum of complexity scores of the parts.

Property 9 ($\neg\text{SUB}$ (W9)) *C is **not subadditive**, i.e. the complexity score of a composed model can be greater than the sum of the complexity of its parts: Let $\mathcal{O} := \{\rightarrow, \wedge, \times, \circlearrowleft\}$.*

$$\forall \oplus \in \mathcal{O} : \exists M_1, M_2 \in \mathcal{M} : C(M_1 \oplus M_2) > C(M_1) + C(M_2)$$

The $\neg$SUB property is especially interesting if a measure does not fulfill it. In this case, we know that the complexity score of a combination of two nets is always less or equal to the sum of complexity scores of the parts.

4.2   Extensions for Weyuker's Properties

In this part, we define properties that extend the ones already defined. A simple but useful property for the complexity of Petri nets was introduced by Morasca [13], who states that a complexity measure should always be defined and return non-negative values. Obviously, this property is also important for complexity measures of process models, since negative complexity scores would be difficult to interpret.

Property 10 (DEF) *The complexity measure C is **defined** for every process model M and returns non-negative values:*

$$\forall M \in \mathcal{M} : C(M) \in \mathbb{R}_0^+$$

Often, we want a complexity measure to have a minimum value to avoid generating models with peculiar structures that lower the complexity. For example, if we want to lower the density of a model, we can add a long chain of τ-transitions after the initial place. A complexity measure not having a minimum value is a hint that this can happen.

Property 11 (MIN) *The complexity measure C has a **minimum** that can be reached by a process model:*

$$\exists m \in \mathcal{M} : \forall M \in \mathcal{M} : C(m) \leq C(M)$$

The last two properties we want to investigate are related to the ¬SUB property. If a complexity measure fulfills ¬SUB it makes sense to analyze it for superadditivity. While subadditive complexity measures can help in estimating the complexity of a composed net, superadditive measures imply that one should take care when using the operations of Definition 6.

Property 12 (¬SUP) *C is **not superadditive**, i.e. the complexity score of a composed model can be less than the sum of the complexity scores of its parts: Let $\mathcal{O} := \{\rightarrow, \wedge, \times, \circlearrowleft\}$.*

$$\forall \oplus \in \mathcal{O} : \exists M_1, M_2 \in \mathcal{M} : C(M_1 \oplus M_2) < C(M_1) + C(M_2)$$

Similarly, if a complexity measure fulfills neither ¬SUB nor ¬SUP, it is interesting to investigate whether this is because the measure is additive. Additive measures remove the need to recalculate the complexity for composed nets.

Property 13 (ADD) *C is **additive**, i.e. the complexity score of a composed model is exactly the sum of the complexity scores of its composed parts: Let $\mathcal{O} := \{\rightarrow, \wedge, \times, \circlearrowleft\}$.*

$$\forall \oplus \in \mathcal{O} : \forall M_1, M_2 \in \mathcal{M} : C(M_1 \oplus M_2) = C(M_1) + C(M_2)$$

Of course, one could imagine more properties for complexity measures. We focus especially on Weyuker's properties and their extensions, since they are well-known for the analysis of software complexity measures and because they were already successfully used on a complexity measure for process models.

5   Analysis of Complexity Measures

In this section, we analyze the complexity measures that were categorized into dimensions by Lieben et al. [9] regarding the properties defined in Section 4. We structure our analyses by the complexity dimension the complexity measures belong to. For each measure, we first give a brief introduction in how the measure works and translate its definition for BPMN into a definition for workflow nets. Then, we go through all properties of Section 4 and give a short indicator whether the measure fulfills the property ($\checkmark$) or not ($\times$), followed by a detailed justification for that claim. Note that we repeat some example nets for different complexity measures. This is because the analyses are intended to work like a dictionary, where the analyses of a complexity measure can be understood without reading the other analyses.

Before starting with the analyses, we show a Theorem that will prove useful for measures that are based on the connectors of a workflow net.

Theorem 1. *Connector metric finitude Let C be a complexity measure and $\mathrm{ran}(C)$ be the set of complexity scores C can return. If C takes only connectors as an input, infinitely many workflow nets can be constructed for each value $c \in \mathrm{ran}(C)$.*

Proof. Let C be a complexity measure, $c \in \text{ran}(C)$ and W_c be a workflow net with $C(W_c) = c$. Then we can insert a sequence of places and transitions between the initial place and the first connector without changing the total number of connectors or their degrees. □

Note that if the additional transitions are silent, they also do not change the language of the model.

5.1 Token Behavior Complexity

The TOKEN BEHAVIOR COMPLEXITY dimension [9] contains complexity measures that record the number of consumed and produced tokens by executing activities. This dimension is the largest of the ones found by Lieben et al., containing seven complexity measures in total.

Size The size of a process model is one of the oldest ways to measure its complexity [3, 8, 13]. For BPMN models, Mendling defines this complexity measure as the number of nodes in the model [3, p.118]. There are variations of this measure that take the number of arcs into account or give the transitions weights according to the complexity of the activity that it represents [13]. While these kinds of measures are also important, we directly adapt the measure of Mendling. The analyes for the aforementioned variations are similar to the ones we perform here. Let $W = (P, T, F, \ell, p_i, p_o)$ be an arbitrary workflow net.

$$C_{\text{size}}(W) = |P| + |T| \tag{1}$$

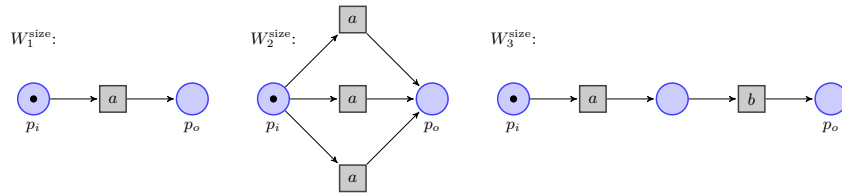Figure 4 shows some simple example nets and their respective complexity score.

Fig. 4: Three simple workflow nets, $W_1^{\text{size}}, W_2^{\text{size}}, W_3^{\text{size}}$, with complexity scores $C_{\text{size}}(W_1^{\text{size}}) = 3$ and $C_{\text{size}}(W_2^{\text{size}}) = C_{\text{size}}(W_3^{\text{size}}) = 5$.

¬**TRIV** (✓): For the two nets W_1^{size} and W_2^{size} of Figure 4, we get the complexity scores $C_{\text{size}}(W_1^{\text{size}}) = 3 \neq 5 = C_{\text{size}}(W_2^{\text{size}})$.

FIN ($\checkmark$): Let $c \in \mathbb{R}$. There are only finitely many workflow nets with complexity c: Suppose we have a directed graph $G = (V, E)$ with $|V| = c$. Then, $|E| \leq c^2$, since all nodes may be connected to all other nodes. If we want to choose only some of these edges, we get $\sum_{i=0}^{c^2} \binom{c^2}{i} = 2^{c^2}$ possible edge sets E. Not all graphs that we can construct in this way can be converted to workflow nets, but every workflow net of size c can be interpreted as such a graph. Therefore, $|\{M \in \mathcal{M} \mid C_{\text{size}}(M) = c\}| < 2^{c^2} < \infty$.

COL ($\checkmark$): The workflow nets W_2^{size} and W_3^{size} of Figure 4 are different in structure, but both get the complexity score $C_{\text{size}}(W_2^{\text{size}}) = 5 = C_{\text{size}}(W_3^{\text{size}})$.

IND$_L$ ($\checkmark$): Take the workflow nets W_1^{size} and W_2^{size} of Figure 4. Their languages are $L(W_1^{\text{size}}) = \{\varepsilon, a\} = L(W_2^{\text{size}})$, but their complexity scores are $C_{\text{size}}(W_1^{\text{size}}) = 3 \neq 5 = C_{\text{size}}(W_2^{\text{size}})$.

MON ($\checkmark$): For this claim, we use the following Theorem:

Theorem 2. *Let $n \geq 2$. For any workflow nets $M_1, \ldots, M_n \in \mathcal{M}$, we get:*
- *$C_{size}(\rightarrow (M_1, \ldots, M_n)) = C_{size}(M_1) + \cdots + C_{size}(M_n) + n - 1$,*
- *$C_{size}(\wedge(M_1, \ldots, M_n)) = C_{size}(M_1) + \cdots + C_{size}(M_n) + 4$,*
- *$C_{size}(\times(M_1, \ldots, M_n)) = C_{size}(M_1) + \cdots + C_{size}(M_n) + 2n + 2$ and*
- *$C_{size}(\circlearrowleft (M_1, \ldots, M_n)) = C_{size}(M_1) + \cdots + C_{size}(M_n) + 2n + 6$.*

Proof. Follows directly from Definition 6. $\qquad\square$

With Theorem 2, we immediately get:
- $C_{\text{size}}(M_1 \rightarrow M_2) = C_{\text{size}}(M_1) + C_{\text{size}}(M_2) + n - 1 > C_{\text{size}}(M_1)$,
- $C_{\text{size}}(M_1 \wedge M_2) = C_{\text{size}}(M_1) + C_{\text{size}}(M_2) + 4 > C_{\text{size}}(M_1)$,
- $C_{\text{size}}(M_1 \times M_2) = C_{\text{size}}(M_1) + C_{\text{size}}(M_2) + 2n + 2 > C_{\text{size}}(M_1)$ and
- $C_{\text{size}}(M_1 \circlearrowleft M_2) = C_{\text{size}}(M_1) + C_{\text{size}}(M_2) + 2n + 6 > C_{\text{size}}(M_1)$.

So C_{size} is, in fact, monotone for all operations of Definition 6.

COMP ($\times$): Let $M_1, M_2, M_3 \in \mathcal{M}$ be three arbitrary workflow nets, where $C_{\text{size}}(M_1) = C_{\text{size}}(M_2)$. Theorem 2 gives:
- $C_{\text{size}}(M_1 \rightarrow M_3) = C_{\text{size}}(M_1) + C_{\text{size}}(M_3) + 1$
$\qquad\qquad = C_{\text{size}}(M_2) + C_{\text{size}}(M_3) + 1 = C_{\text{size}}(M_2 \rightarrow M_3)$,
- $C_{\text{size}}(M_1 \wedge M_3) = C_{\text{size}}(M_1) + C_{\text{size}}(M_3) + 4$
$\qquad\qquad = C_{\text{size}}(M_2) + C_{\text{size}}(M_3) + 4 = C_{\text{size}}(M_2 \wedge M_3)$,
- $C_{\text{size}}(M_1 \times M_3) = C_{\text{size}}(M_1) + C_{\text{size}}(M_3) + 6$
$\qquad\qquad = C_{\text{size}}(M_2) + C_{\text{size}}(M_3) + 6 = C_{\text{size}}(M_2 \times M_3)$ and
- $C_{\text{size}}(M_1 \circlearrowleft M_3) = C_{\text{size}}(M_1) + C_{\text{size}}(M_3) + 10$
$\qquad\qquad = C_{\text{size}}(M_2) + C_{\text{size}}(M_3) + 10 = C_{\text{size}}(M_2 \circlearrowleft M_3)$.

Thus, none of the operations of Definition 6 are sensitive to compositions.

PERM ($\checkmark$): According to our definition of permutations, the net W_4^{size} of Figure 5 is a permutation of the net W_3^{size} of Figure 4, but their complexity scores are $C_{\text{size}}(W_3^{\text{size}}) = 5 \neq 6 = C_{\text{size}}(W_4^{C_{\text{size}}})$.

ROB$_\ell$ ($\checkmark$): The labeling of the transitions doesn't influence the number of places and transitions in a workflow net.

¬SUB ($\checkmark$): According to Theorem 2, we can take any $M_1, M_2 \in \mathcal{M}$, as well as any $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$ and get $C_{\text{size}}(M_1 \oplus M_2) > C_{\text{size}}(M_1) + C_{\text{size}}(M_2)$.
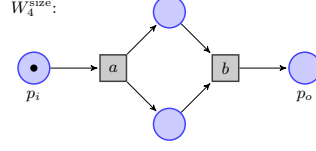
Fig. 5: A workflow net W_4^{size} with complexity score $C_{\text{size}}(W_4^{\text{size}}) = 6$.

DEF ($\checkmark$): C_{size} is defined for all workflow nets $M = (P, T, F, p_i, p_o) \in \mathcal{M}$. By definition, $|P|$ and $|T|$ are non-negative values, so $C_{\text{size}}(M) \geq 0$.

MIN ($\checkmark$): By definition, the workflow net with the smallest amount of nodes is the net W_0 of Figure 2. So $C_{\text{size}}(W_0) = 3$ is the minimum score of this complexity metric.

INF ($\checkmark$): For every $c \in \mathbb{N}$ with $c \geq 3$, we can find a workflow net with complexity c by introducing two places p_i and p_o, as well as $c-2$ transitions that have an edge coming from p_i and one leading to p_o. Therefore, we get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\text{size}}(M) = c\}| \geq \{c \in \mathbb{N} \mid c \geq 3\}| = \infty$.

¬SUP ($\times$): Follows directly from Theorem 2.

ADD ($\times$): Follows directly from Theorem 2.

Connector Mismatch A connector mismatch happens when a split connector of one type is paired with a join connector of another type. Such mismatches lead to undesired behavior for the tokens in the net. Furthermore, since humans tend to prefer symmetry, such a structure is difficult to understand and indicates errors in the model [3]. Analyzing how many mismatches occur in a workflow net is not trivial, as it requires analyzing all paths starting from a split-node. In a graph, with n vertices, the number of paths can be exponential in n, so this would be a time-intensive analysis.

To approximate the amount of connector mismatches, Mendling [3, p.125] defines the number of mismatches of a type as the difference of arcs leaving a split node and arcs entering a join node of that type. We adapt this definition for workflow nets by defining, for $W = (P, T, F, \ell, p_i, p_o) \in \mathcal{M}$,

$$MM_{\text{and}}^W := \left| \sum_{t \in \mathcal{S}_{\text{and}}^W} |t^\bullet| - \sum_{t \in \mathcal{J}_{\text{and}}^W} |^\bullet t| \right| \tag{2}$$

as the mismatch between concurrent split and join connectors and

$$MM_{\text{xor}}^W := \left| \sum_{p \in \mathcal{S}_{\text{xor}}^W} |t^\bullet| - \sum_{p \in \mathcal{J}_{\text{xor}}^W} |^\bullet p| \right| \tag{3}$$

as the mismatch between xor split and join connectors. According to the measure defined by Mendling, the total amount of mismatches in the model is defined as the sum of these mismatches. For workflow nets, we therefore define

$$C_{\text{MM}}(W) = MM_{\text{and}}^W + MM_{\text{xor}}^W. \tag{4}$$

The original definition also takes or-connectors into account. Since we don't have designated connectors for inclusive choices in workflow nets, we ignore this part of the complexity measure. Figure 6 shows some simple example nets and their respective complexity score.
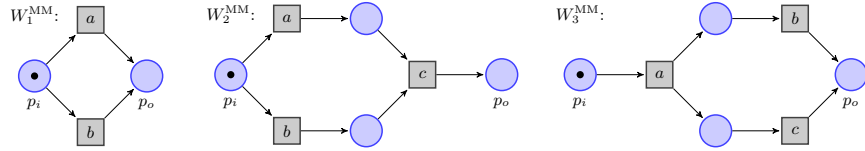

Fig. 6: Three simple workflow nets, $W_1^{\mathrm{MM}}, W_2^{\mathrm{MM}}, W_3^{\mathrm{MM}}$, with complexity scores $C_{\mathrm{MM}}(W_1^{\mathrm{MM}}) = 0$ and $C_{\mathrm{MM}}(W_2^{\mathrm{MM}}) = C_{\mathrm{MM}}(W_3^{\mathrm{MM}}) = 4$.

$\neg$**TRIV** ($\checkmark$): For the two nets W_1^{MM} and W_2^{MM} of Figure 6, we get the complexity scores $C_{\mathrm{MM}}(W_1^{\mathrm{MM}}) = 0 \neq 4 = C_{\mathrm{MM}}(W_2^{\mathrm{MM}})$.

FIN ($\times$): Follows directly from Theorem 1. Figure 7 shows how to construct such a sequence of places and transitions.
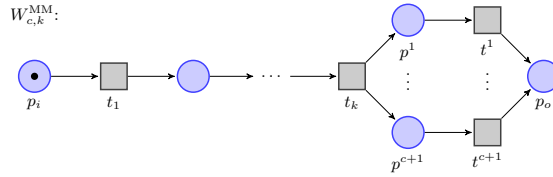

Fig. 7: A workflow net $W_{c,k}^{\mathrm{MM}}$ with $k + c + 1$ transitions and complexity score $C_{\mathrm{MM}}(W_{c,k}^{\mathrm{MM}}) = 2 \cdot (c + 1)$.

COL ($\checkmark$): The workflow nets W_2^{MM} and W_3^{MM} of Figure 6 are different in structure, but both get the complexity score $C_{\mathrm{MM}}(W_2^{\mathrm{MM}}) = 4 = C_{\mathrm{MM}}(W_3^{\mathrm{MM}})$.

IND$_L$ ($\checkmark$): Take the workflow nets W_1^{MM} and W_2^{MM} of Figure 6. Their languages are $L(W_1^{\mathrm{MM}}) = \{\varepsilon, a, b\} = L(W_2^{\mathrm{MM}})$, but their complexity scores are $C_{\mathrm{MM}}(W_1^{\mathrm{MM}}) = 0 \neq 4 = C_{\mathrm{MM}}(W_2^{\mathrm{MM}})$.

MON ($\times$): Take the workflow nets W_2^{MM} and W_3^{MM} of Figure 6, as well as any operation $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowright\}$. Each operation introduces a split-connector of exactly one type, as well as a join-connector of the same type. The number of outgoing arcs of the split-connector are exactly the number of incoming arcs of the join-connector. However, W_2^{MM} introduces 2 arcs leaving an and-connector and 2 arcs entering an xor-connector, while W_3^{MM} introduces 2 arcs leaving an xor-connector and 2 arcs entering an and-connector. In total, we therefore get $C_{\mathrm{MM}}(W_2^{\mathrm{MM}} \oplus W_3^{\mathrm{MM}}) = 0 < 4 = C_{\mathrm{MM}}(W_2^{\mathrm{MM}})$.

COMP ($\checkmark$): Take the workflow nets W_2^{MM} and W_3^{MM} of Figure 6. As the caption of this figure suggests, $C_{\mathrm{MM}}(W_2^{\mathrm{MM}}) = C_{\mathrm{MM}}(W_3^{\mathrm{MM}})$, but if we combine these nets with any of the operations $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$ with W_3^{MM}, we get $C_{\mathrm{MM}}(W_2^{\mathrm{MM}} \oplus W_3^{\mathrm{MM}}) = 0 \neq 8 = C_{\mathrm{MM}}(W_3^{\mathrm{MM}} \oplus W_3^{\mathrm{MM}})$.

PERM ($\checkmark$): Figure 8 shows two nets with different complexity scores according to C_{MM} that are permutations of each other.
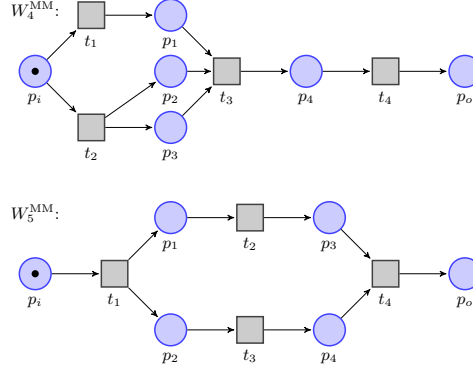

Fig. 8: Two workflow nets, W_4^{MM} and W_5^{MM}, where $W_4^{\mathrm{MM}} \in Perm(W_5^{\mathrm{MM}})$. W_5^{MM} contains no connector mismatches, so $C_{\mathrm{MM}}(W_5^{\mathrm{MM}}) = 0$, while $C_{\mathrm{MM}}(W_4^{\mathrm{MM}}) = 2$.

ROB$_\ell$ ($\checkmark$): C_{MM} depends only on the connector types in the net and their out- and in-degree. However, we cannot change these by performing a relabeling on the net.

$\neg$**SUB** ($\times$): Let $M_1 = (P_1, T_1, F_1, p_i^1, p_o^1)$ and $M_2 = (P_2, T_2, F_2, p_i^2, p_o^2)$ be workflow nets and let $M = M_1 \oplus M_2$ for some $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$. We get:

$$
\begin{aligned}
MM_{\mathrm{and}}^M &= \left| \sum_{t \in \mathcal{S}_{\mathrm{and}}^M} |t^\bullet| - \sum_{t \in \mathcal{J}_{\mathrm{and}}^M} |^\bullet t| \right| \\
&= \left| \sum_{t \in \mathcal{S}_{\mathrm{and}}^{M_1} \cup \mathcal{S}_{\mathrm{and}}^{M_2}} |t^\bullet| - \sum_{t \in \mathcal{J}_{\mathrm{and}}^{M_1} \cup \mathcal{J}_{\mathrm{and}}^{M_2}} |^\bullet t| \right| \\
&\leq \left| \sum_{t \in \mathcal{S}_{\mathrm{and}}^{M_1}} |t^\bullet| - \sum_{t \in \mathcal{J}_{\mathrm{and}}^{M_1}} |^\bullet t| \right| + \left| \sum_{t \in \mathcal{S}_{\mathrm{and}}^{M_2}} |t^\bullet| - \sum_{t \in \mathcal{J}_{\mathrm{and}}^{M_2}} |^\bullet t| \right| \\
&= MM_{\mathrm{and}}^{M_1} + MM_{\mathrm{and}}^{M_2}
\end{aligned}
$$

where the second equality is due to the fact that $T_1 \cap T_2 = \emptyset$ by definition. Analogously, we get $MM_{\text{xor}}^M \leq MM_{\text{xor}}^{M_1} + MM_{\text{xor}}^{M_2}$ and therefore

$$
\begin{aligned}
C_{\text{MM}}(W) &= MM_{\text{and}}^M + MM_{\text{xor}}^M \\
&\leq (MM_{\text{and}}^{M_1} + MM_{\text{and}}^{M_2}) + (MM_{\text{xor}}^{M_1} + MM_{\text{xor}}^{M_2}) \\
&= (MM_{\text{and}}^{M_1} + MM_{\text{xor}}^{M_1}) + (MM_{\text{and}}^{M_2} + MM_{\text{xor}}^{M_2}) \\
&= C_{\text{MM}}(M_1) + C_{\text{MM}}(M_2).
\end{aligned}
$$

DEF ($\checkmark$): Since $\mathcal{S}_{\text{and}}^W, \mathcal{S}_{\text{xor}}^W, \mathcal{J}_{\text{and}}^W, \mathcal{J}_{\text{xor}}^W$ are defined for all workflow nets $W \in \mathcal{M}$, $C_{\text{MM}}(W)$ is also defined for all such workflow nets. Furthermore, since we sum only over absolute values to compute $C_{\text{MM}}(W)$, we get $C_{\text{MM}}(W) \geq 0$ for all workflow nets $W \in \mathcal{M}$.

MIN ($\checkmark$): The minimum for this metric is 0. A workflow net without any connectors, like W_0 of Figure 2, receives this complexity score.

INF ($\checkmark$): For any $c \in \mathbb{N}$, Figure 7 shows how to construct a workflow net with complexity score $2c + 2$. Therefore, there are infinitely many possible complexity scores:

$|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\text{MM}}(M) = c\}| \geq |\{n \in \mathbb{N} \mid \exists c \in \mathbb{N} : n = 2c + 2\}| = \infty.$

¬SUP ($\checkmark$): Consider again the workflow nets W_2^{MM} and W_3^{MM} in Figure 6. Since none of the operations $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$ introduce new connector mismatches, we get $C_{\text{MM}}(W_2^{\text{MM}} \oplus W_3^{\text{MM}}) = 0 < 8 = C_{\text{MM}}(W_2^{\text{MM}}) + C_{\text{MM}}(W_3^{\text{MM}})$.

ADD ($\times$): We can use the same counter-example as for the property ¬SUP to see that C_{MM} is not additive.

Connector Heterogeneity According to Mendling [3, pp.126-127], an EPC model that contains many connector types is more prone to errors, since it is more likely to contain a connector mismatch. He therefore introduces the connector heterogeneity metric, which calculates the entropy of connectors in the model. Since we don't have designated connectors for inclusive choices, we only calculate the entropy of concurrent and exclusive choice connectors. Let $W = (P, T, F, \ell, p_i, p_o)$ be a workflow net.

$$
C_{\text{CH}}(W) = - \left(\frac{|\mathcal{C}_{\text{and}}^W|}{|\mathcal{C}^W|} \cdot \log_2 \left(\frac{|\mathcal{C}_{\text{and}}^W|}{|\mathcal{C}^W|} \right) + \frac{|\mathcal{C}_{\text{xor}}^W|}{|\mathcal{C}^W|} \cdot \log_2 \left(\frac{|\mathcal{C}_{\text{xor}}^W|}{|\mathcal{C}^W|} \right) \right) \qquad (5)
$$

A value near 0 means that there are only connectors of one type present in the net, while a value near 1 means that we have equally many parallel and choice connectors. Figure 9 shows some simple example nets and their respective complexity score.

¬TRIV ($\checkmark$): For the two nets W_1^{CH} and W_2^{CH} of Figure 9, we get the complexity scores $C_{\text{CH}}(W_1^{\text{CH}}) \approx 0.9852 \neq 0 = C_{\text{CH}}(W_2^{\text{CH}})$.

FIN ($\times$): Follows directly from Theorem 1. Figure 10 shows how to construct such a sequence of places and transitions.

COL ($\checkmark$): The workflow nets W_2^{CH} and W_3^{CH} of Figure 9 are different in structure, but both get the complexity score $C_{\text{CH}}(W_2^{\text{CH}}) = 0 = C_{\text{CH}}(W_3^{\text{CH}})$.
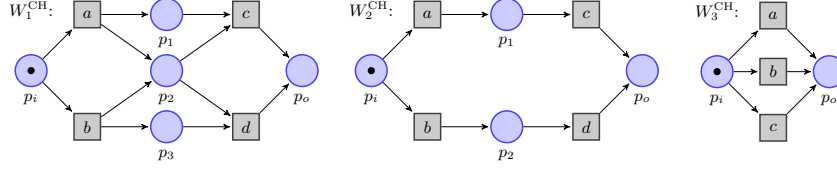
Fig. 9: Workflow nets, $W_1^{\mathrm{CH}}, W_2^{\mathrm{CH}}, W_3^{\mathrm{CH}}$, with $C_{\mathrm{CH}}(W_2^{\mathrm{CH}}) = C_{\mathrm{CH}}(W_3^{\mathrm{CH}}) = 0$ and $C_{\mathrm{CH}}(W_1^{\mathrm{CH}}) = -(\frac{4}{7} \cdot \log_2(\frac{4}{7}) + \frac{3}{7} \cdot \log_2(\frac{3}{7})) \approx 0.9852$.
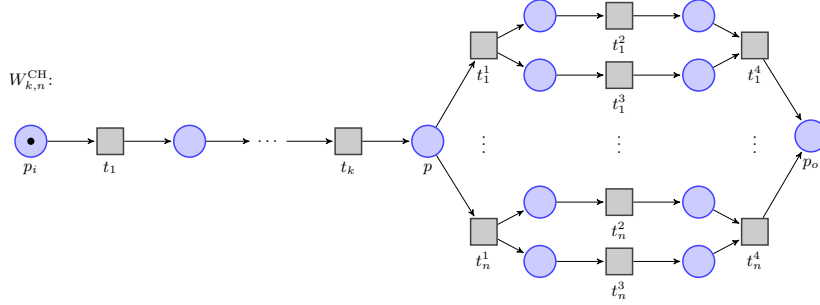

Fig. 10: A construction plan for workflow nets $W_{k,n}^{\mathrm{CH}}$ with $k + 4n$ transitions and $C_{\mathrm{CH}}(W_{k,n}^{\mathrm{CH}}) = -\left(\frac{1}{n+1} \cdot \log_2(\frac{1}{n+1}) + \frac{n}{n+1} \cdot \log_2(\frac{n}{n+1}) \right)$ for $k, n \in \mathbb{N}$ with $n \geq 2$.

IND$_L$ ($\checkmark$): Take the workflow nets W_1^{CH} and W_2^{CH} of Figure 9. Their languages are $L(W_1^{\mathrm{CH}}) = \{\varepsilon, a, b, ac, bd\} = L(W_2^{\mathrm{CH}})$, but their complexity scores are $C_{\mathrm{CH}}(W_1^{\mathrm{CH}}) = 0.9852 \neq 0 = C_{\mathrm{CH}}(W_2^{\mathrm{CH}})$.

MON ($\times$): The measure C_{CH} does not fulfill this property for any of the operations $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$. To construct counter examples, consider the workflow net W_1^{CH} of Figure 9 and the workflow net $W_{1,3}^{\mathrm{CH}}$ of Figure 10.

- $C_{\mathrm{CH}}(W_1^{\mathrm{CH}} \rightarrow W_{1,3}^{\mathrm{CH}}) = -\left(\frac{10}{12} \cdot \log_2(\frac{10}{12}) + \frac{2}{12} \cdot \log_2(\frac{2}{12}) \right) \approx 0.6500$
  $< 0.9852 \approx C_{\mathrm{CH}}(W_1^{\mathrm{CH}})$,
- $C_{\mathrm{CH}}(W_1^{\mathrm{CH}} \wedge W_{1,3}^{\mathrm{CH}}) = -\left(\frac{12}{14} \cdot \log_2(\frac{12}{14}) + \frac{2}{14} \cdot \log_2(\frac{2}{14}) \right) \approx 0.5917$
  $< 0.9852 \approx C_{\mathrm{CH}}(W_1^{\mathrm{CH}})$,
- $C_{\mathrm{CH}}(W_1^{\mathrm{CH}} \times W_{1,3}^{\mathrm{CH}}) = -\left(\frac{10}{14} \cdot \log_2(\frac{10}{14}) + \frac{4}{14} \cdot \log_2(\frac{4}{14}) \right) \approx 0.8631$
  $< 0.9852 \approx C_{\mathrm{CH}}(W_1^{\mathrm{CH}})$,
- $C_{\mathrm{CH}}(W_1^{\mathrm{CH}} \circlearrowleft W_1^{\mathrm{CH}}) = -\left(\frac{10}{14} \cdot \log_2(\frac{10}{14}) + \frac{4}{14} \cdot \log_2(\frac{4}{14}) \right) \approx 0.8631$
  $< 0.9852 \approx C_{\mathrm{CH}}(W_1^{\mathrm{CH}})$.

COMP ($\checkmark$): Consider the two workflow nets W_4^{CH} and W_5^{CH} of Figure 11 and the workflow net W_3^{CH} of Figure 9. For their complexity scores, we have $C_{\mathrm{CH}}(W_4^{\mathrm{CH}}) = 0.9183 = C_{\mathrm{CH}}(W_5^{\mathrm{CH}})$, but for the operations of Definition 6, we get:
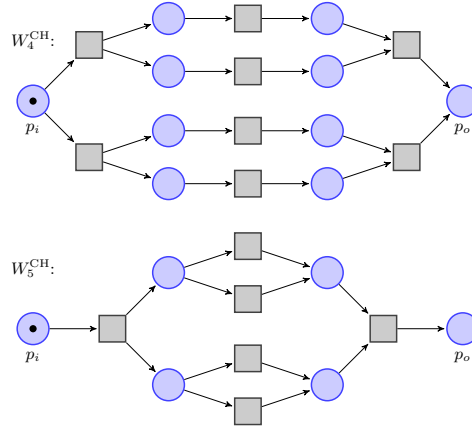
Fig. 11: Two workflow nets with different structure but the same complexity score $C_{\mathrm{CH}}(W_4^{\mathrm{CH}}) = C_{\mathrm{CH}}(W_5^{\mathrm{CH}}) = -\left(\frac{2}{6} \cdot \log_2(\frac{2}{6}) + \frac{4}{6} \cdot \log_2(\frac{4}{6})\right) \approx 0.9183$.

- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \to W_3^{\mathrm{CH}}) = -\left(\frac{4}{8} \cdot \log_2(\frac{4}{8}) + \frac{4}{8} \cdot \log_2(\frac{4}{8})\right) = 0$ and
  $C_{\mathrm{CH}}(W_5^{\mathrm{CH}} \to W_3^{\mathrm{CH}}) = -\left(\frac{2}{8} \cdot \log_2(\frac{2}{8}) + \frac{6}{8} \cdot \log_2(\frac{6}{8})\right) \approx 0.8113$,
- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \wedge W_3^{\mathrm{CH}}) = -\left(\frac{5}{9} \cdot \log_2(\frac{5}{9}) + \frac{4}{9} \cdot \log_2(\frac{4}{9})\right) \approx 0.9911$ and
  $C_{\mathrm{CH}}(W_5^{\mathrm{CH}} \wedge W_3^{\mathrm{CH}}) = -\left(\frac{3}{9} \cdot \log_2(\frac{3}{9}) + \frac{6}{9} \cdot \log_2(\frac{6}{9})\right) \approx 0.9183$,
- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \times W_3^{\mathrm{CH}}) = -\left(\frac{4}{9} \cdot \log_2(\frac{4}{9}) + \frac{5}{9} \cdot \log_2(\frac{5}{9})\right) \approx 0.9911$ and
  $C_{\mathrm{CH}}(W_5^{\mathrm{CH}} \times W_3^{\mathrm{CH}}) = -\left(\frac{2}{9} \cdot \log_2(\frac{2}{9}) + \frac{7}{9} \cdot \log_2(\frac{7}{9})\right) \approx 0.7642$,
- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \circlearrowright W_3^{\mathrm{CH}}) = -\left(\frac{4}{9} \cdot \log_2(\frac{4}{9}) + \frac{5}{9} \cdot \log_2(\frac{5}{9})\right) \approx 0.9911$ and
  $C_{\mathrm{CH}}(W_5^{\mathrm{CH}} \circlearrowright W_3^{\mathrm{CH}}) = -\left(\frac{2}{9} \cdot \log_2(\frac{2}{9}) + \frac{7}{9} \cdot \log_2(\frac{7}{9})\right) \approx 0.7642$.

PERM (✓): The workflow nets W_6^{CH} and W_7^{CH} of Figure 12 are permutations of each other, but get different complexity scores:

$$C_{\mathrm{CH}}(W_6^{\mathrm{CH}}) = -\left(\frac{2}{4} \cdot \log_2\left(\frac{2}{4}\right) + \frac{2}{4} \cdot \log_2\left(\frac{2}{4}\right)\right) = 1$$

$$\neq 0.9182 \approx -\left(\frac{4}{6} \cdot \log_2\left(\frac{4}{6}\right) + \frac{2}{6} \cdot \log_2\left(\frac{2}{6}\right)\right) = C_{\mathrm{CH}}(W_7^{\mathrm{CH}})$$

ROB$_\ell$ (✓): C_{CH} depends only on the amount of connectors of each type, which is independent of the labeling.

¬**SUB** (✓): Take the workflow nets W_4^{CH} and W_5^{CH} of Figure 11. Both of these workflow nets have complexity 0 according to C_{CH}, but combining them using any of the operations $\to, \wedge, \times, \circlearrowright$ yields:

- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \to W_5^{\mathrm{CH}}) = -\left(\frac{6}{12} \cdot \log_2(\frac{6}{12}) + \frac{6}{12} \cdot \log_2(\frac{6}{12})\right)$
  $= 1 > 0 + 0 = C_{\mathrm{CH}}(W_4^{\mathrm{CH}}) + C_{\mathrm{CH}}(W_5^{\mathrm{CH}})$,
- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \wedge W_5^{\mathrm{CH}}) = -\left(\frac{8}{14} \cdot \log_2(\frac{8}{14}) + \frac{6}{14} \cdot \log_2(\frac{6}{14})\right)$
  $\approx 0.9852 > 0 + 0 = C_{\mathrm{CH}}(W_4^{\mathrm{CH}}) + C_{\mathrm{CH}}(W_5^{\mathrm{CH}})$,
- $C_{\mathrm{CH}}(W_4^{\mathrm{CH}} \times W_5^{\mathrm{CH}}) = -\left(\frac{6}{14} \cdot \log_2(\frac{6}{14}) + \frac{8}{14} \cdot \log_2(\frac{8}{14})\right)$
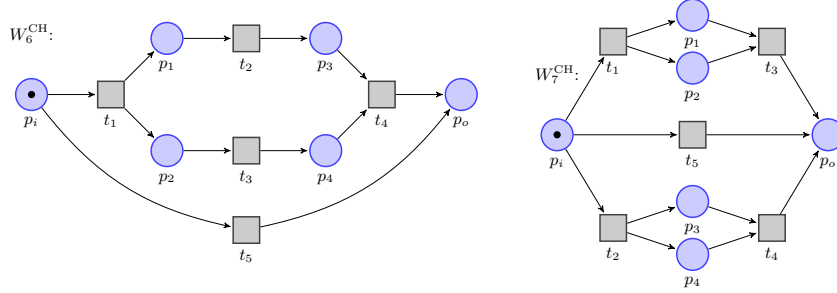 $\approx 0.9852 > 0 + 0 = C_{\mathrm{CH}}(W_4^{\mathrm{CH}}) + C_{\mathrm{CH}}(W_5^{\mathrm{CH}})$,

Fig. 12: Two workflow nets, W_6^{CH} and W_7^{CH}, where $W_7^{\text{CH}} \in Perm(W_6^{\text{CH}})$, but $C_{\text{CH}}(W_6^{name}) \neq C_{\text{CH}}(W_7^{\text{CH}})$.

$$- \; C_{\text{CH}}(W_4^{\text{CH}} \circlearrowleft W_5^{\text{CH}}) = - \left(\tfrac{6}{14} \cdot \log_2(\tfrac{6}{14}) + \tfrac{8}{14} \cdot \log_2(\tfrac{8}{14}) \right)$$
$$\approx 0.9852 > 0 + 0 = C_{\text{CH}}(W_4^{\text{CH}}) + C_{\text{CH}}(W_5^{\text{CH}}).$$

DEF ($\times$): The connector heterogeneity metric is undefined for workflow nets without connectors, since we would divide by 0 in this case. We can avoid this by defining that C_{CH} should be 0 in this special case, since a workflow net without any connectors is sequential and therefore easy to understand. In the defined cases, the metric only returns non-negative values, since $\log_2(r) < 0$ for some $0 < r \leq 1$. Adding two non-positive values results in a non-positive value, so negating this value results in a non-negative value.

MIN ($\checkmark$): The minimum possible complexity score of C_{CH} is 0. This score is rewarded to a workflow net that contains only one connector type, like the net W_2^{CH} in Figure 9, which contains only xor-connectors.

INF ($\checkmark$): Let $k, n \in \mathbb{N}$ with $n \geq 2$ be fixed. Figure 10 shows how to construct a workflow net with 2 xor-connectors (the places p and p_o) and $2n$ and-connectors (the transitions $t_1^1, \ldots t_n^1, t_1^4, \ldots, t_n^4$). We therefore get

$$C_{\text{CH}}(W_{k,n}^{\text{CH}}) = - \left(\frac{1}{n+1} \cdot \log_2 \left(\frac{1}{n+1} \right) + \frac{n}{n+1} \cdot \log_2 \left(\frac{n}{n+1} \right) \right)$$
$$> - \left(\frac{1}{n+2} \cdot \log_2 \left(\frac{1}{n+2} \right) + \frac{n+1}{n+2} \cdot \log_2 \left(\frac{n+1}{n+2} \right) \right)$$
$$= C_{\text{CH}}(W_{k,n+1}^{\text{CH}})$$

Thus, $C_{\text{CH}}(W_{k,n}^{\text{CH}}) \neq C_{\text{CH}}(W_{k,m}^{\text{CH}})$ for any $n \neq m \in \mathbb{N}$ with $n, m \geq 2$, so Figure 10 shows how to construct infinitely many workflow nets of different complexity according to C_{CH}. In other words, we get the following result: $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\text{CH}}(M) = c\}| \geq |\{n \in \mathbb{N} \mid n \geq 2\}| = \infty$.

¬SUP ($\checkmark$): The counter examples found for property MON also show that C_{CH} is not superadditive.

ADD ($\times$): Since C_{CH} is neither subadditive nor superadditive, it can't be additive.

Cross-Connectivity Vanderfeesten et al. [14] introduced the cross-connectivity metric to measure how difficult it is to understand the connection between two nodes in a model. This measure is based on paths between all pairs of nodes. Let $N = (P, T, F, p_i, p_o)$ be a workflow net (note that we choose N as the identifier here to avoid confusion with the weight-function needed for this metric). First, each node $v \in P \cup T$ of the net gets a weight depending on its type and degree $\deg(v) = |{}^\bullet n| + |n^\bullet|$:

$$w_N(v) := \begin{cases} \frac{1}{\deg(v)} & \text{if } v \in \mathcal{C}_{\text{xor}}^N \\ 1 & \text{if } v \in \mathcal{C}_{\text{and}}^N \\ 1 & \text{otherwise.} \end{cases} \qquad (6)$$

For an edge $(u, v) \in F$, we define its weight as the product of its endpoints, so $w_N((u, v)) := w_N(u) \cdot w_N(v)$. Let $\rho = v_1, \ldots, v_k$ be a path of length $k \geq 2$. The weight of the path ρ is defined as the product of the edges it uses, so $w_N(\rho) = w_N((v_1, v_2)) \cdot \ldots \cdot w_N((v_{k-1}, v_k))$. Let $\mathcal{P}_{v_1, v_2}$ be the set of all paths from v_1 to v_2 of length at least 2. We define the value of a connection between two nodes v_1 and v_2 as

$$V_N(v_1, v_2) := \max(\{w_N(\rho) \mid \rho \in \mathcal{P}_{v_1, v_2}\} \cup \{0\}). \qquad (7)$$

We use the special value 0 for cases where no path from v_1 to v_2 exist. The original work of Vanderfeesten et al. [14] then defines the cross-connectivity metric as the sum of all values between nodes divided by the number of nodes times the number of nodes minus one. We directly translate this definition to workflow nets, but subtract the result from one, so a low complexity score refers to models with low complexity.

$$C_{\text{CC}}(N) = 1 - \frac{\sum_{v_1, v_2 \in P \cup T} V_N(v_1, v_2)}{(|P| + |T|) \cdot (|P| + |T| - 1)}. \qquad (8)$$

Figure 13 shows a composed workflow net and its complexity score, as well as the scores for the input models.

- **¬TRIV** (✓): For the two nets W_1^{CC} and W_2^{CC} of Figure 13, we get the complexity scores $C_{\text{CC}}(W_1^{\text{CC}}) \approx 0.8796 \neq 0.5 = C_{\text{CC}}(W_2^{\text{CC}})$.
- **FIN** (×): Figure 14 shows how to construct infinitely many workflow nets with complexity score $\frac{1}{2}$.
- **COL** (✓): The workflow nets $W_{fin,1}^{\text{CC}}$ and $W_{fin,2}^{\text{CC}}$ of Figure 14 are structurally different but get the same complexity scores.
- **IND$_L$** (✓): The workflow nets W_3^{CC} and W_4^{CC} of Figure 15 have the same language but receive different complexity scores.
- **MON** (×): C_{CC} is not monotone for any of the operations $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$. For the operator $\wedge$, Figure 13 shows a detailed counter-example for monotonicity. We can take the workflow nets W_1^{CC} and W_2^{CC} of this figure to construct counter-examples for the other operations:
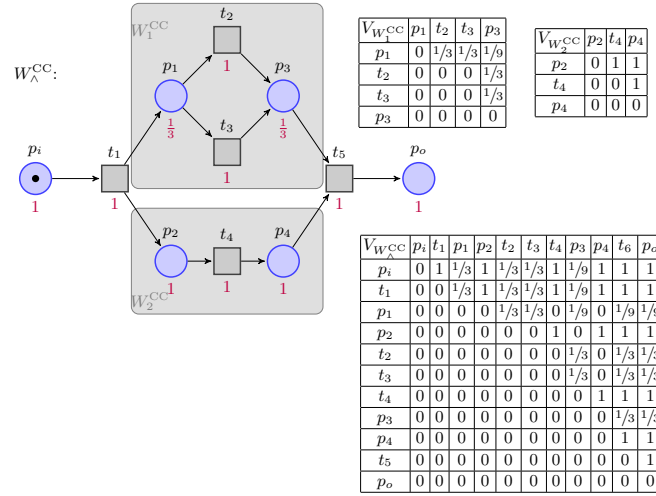
$W_\wedge^{CC}$:

$V_{W_1^{CC}}$	p_1	t_2	t_3	p_3
p_1	0	1/3	1/3	1/9
t_2	0	0	0	1/3
t_3	0	0	0	1/3
p_3	0	0	0	0

$V_{W_2^{CC}}$	p_2	t_4	p_4
p_2	0	1	1
t_4	0	0	1
p_4	0	0	0

$V_{W_\wedge^{CC}}$	p_i	t_1	p_1	p_2	t_2	t_3	t_4	p_3	p_4	t_5	p_o
p_i	0	1	1/3	1	1/3	1/3	1	1/9	1	1	1
t_1	0	0	1/3	1	1/3	1/3	1	1/9	1	1	1
p_1	0	0	0	0	1/3	1/3	0	1/9	0	1/9	1/9
p_2	0	0	0	0	0	0	1	0	1	1	1
t_2	0	0	0	0	0	0	0	1/3	0	1/3	1/3
t_3	0	0	0	0	0	0	0	1/3	0	1/3	1/3
t_4	0	0	0	0	0	0	0	0	1	1	1
p_3	0	0	0	0	0	0	0	0	0	1/3	1/3
p_4	0	0	0	0	0	0	0	0	0	1	1
t_5	0	0	0	0	0	0	0	0	0	0	1
p_o	0	0	0	0	0	0	0	0	0	0	0

Fig. 13: A workflow net $W_\wedge^{CC}$ and its $V_{W_\wedge^{CC}}$-values for all pairs of vertices. We get $C_{CC}(W_\wedge^{CC}) = 1 - \frac{21 + 16\cdot 1/3 + 5\cdot 1/9}{11\cdot 10} = 0.7\overline{5}$, $C_{CC}(W_1^{CC}) = 1 - \frac{4\cdot 1/3 + 1/9}{4\cdot 3} \approx 0.8796$ and $C_{CC}(W_2^{CC}) = 1 - \frac{3}{6} = 0.5$.

- $C_{CC}(W_1^{CC} \to W_2^{CC}) = 1 - \frac{6\cdot 1 + 16\cdot 1/3 + 5\cdot 1/9}{8\cdot 7} \approx 0.7877$
  $< 0.8796 = C_{CC}(W_1^{CC})$,
- $C_{CC}(W_1^{CC} \wedge W_2^{CC}) = 1 - \frac{21\cdot 1 + 16\cdot 1/3 + 5\cdot 1/9}{11\cdot 10} = 0.7\overline{5}$
  $< 0.8796 = C_{CC}(W_1^{CC})$,
- $C_{CC}(W_1^{CC} \times W_2^{CC}) = 1 - \frac{10\cdot 1 + 11\cdot 1/2 + 11\cdot 1/3 + 1/4 + 6\cdot 1/6 + 4\cdot 1/9 + 4\cdot 1/18}{13\cdot 12} \approx 0.8648$
  $< 0.8796 = C_{CC}(W_1^{CC})$,
- $C_{CC}(W_1^{CC} \circlearrowright W_2^{CC}) = 1 - \frac{17 + 37\cdot 1/3 + 56\cdot 1/9 + 43\cdot 1/27 + 74\cdot 1/81}{17\cdot 17} \approx 0.8602$
  $< 0.8796 = C_{CC}(W_1^{CC})$.

COMP (✓): Take the workflow nets $W_{fin,1}^{CC}$ and $W_{fin,2}^{CC}$ of Figure 14. Further, take the workflow nets W_5^{CC} and W_6^{CC} of Figure 16 and the net W_1^{CC} of Figure 13. For the complexity scores, we have $W_{fin,1}^{CC} = W_{fin,2}^{CC} = \frac{1}{2}$ and $W_5^{CC} = W_6^{CC} = 1 - \frac{1\cdot 1 + 2\cdot 1/2 + 8\cdot 1/3 + 3\cdot 1/6}{6\cdot 5} = \frac{151}{180} \approx 0.8389$ and get:

- $C_{CC}(W_5^{CC} \to W_1^{CC}) = 1 - \frac{1\cdot 1 + 2\cdot 1/2 + 18\cdot 1/3 + 2\cdot 1/6 + 15\cdot 1/9 + 9\cdot 1/27 + 3\cdot 1/54}{11\cdot 10} = \frac{163}{180}$
  $\approx 0.9056 \neq 0.8634 \approx \frac{3419}{3960} = 1 - \frac{3\cdot 1 + 4\cdot 1/2 + 22\cdot 1/3 + 8\cdot 1/6 + 9\cdot 1/9 + 6\cdot 1/18 + 1\cdot 1/36}{11\cdot 10}$
  $= C_{CC}(W_6^{CC} \to W_1^{CC})$,
- $C_{CC}(W_{fin,1} \wedge W_{fin,1}) = 1 - \frac{36\cdot 1}{10\cdot 9} = \frac{3}{5} = 0.6$
  $\neq 0.6136 \approx \frac{27}{44} = 1 - \frac{51\cdot 1}{12\cdot 11} = C_{CC}(W_{fin,2} \wedge W_{fin,1})$,
- $C_{CC}(W_{fin,1} \times W_{fin,1}) = 1 - \frac{20\cdot 1 + 20\cdot 1/2 + 1\cdot 1/4}{12\cdot 11} = \frac{37}{48} \approx 0.7708$
  $\neq 0.7624 \approx \frac{555}{728} = 1 - \frac{31\cdot 1 + 24\cdot 1/2 + 1\cdot 1/4}{14\cdot 13} = C_{CC}(W_{fin,2} \times W_{fin,1})$,
- $C_{CC}(W_{fin,1} \circlearrowright W_{fin,1}) = 1 - \frac{22\cdot 1 + 94\cdot 1/3 + 80\cdot 1/9 + 2\cdot 1/27}{16\cdot 15} = \frac{2399}{3240} \approx 0.7404$
  $\neq 0.7194 \approx \frac{2972}{4131} = 1 - \frac{33\cdot 1 + 126\cdot 1/3 + 97\cdot 1/9 + 2\cdot 1/27}{18\cdot 17} = C_{CC}(W_{fin,2} \circlearrowright W_{fin,1})$.
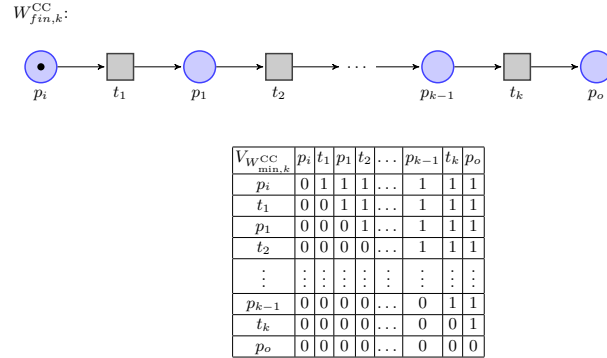
$W_{fin,k}^{\mathrm{CC}}$:

$V_{W_{\min,k}^{\mathrm{CC}}}$	p_i	t_1	p_1	t_2	$\ldots$	p_{k-1}	t_k	p_o
p_i	0	1	1	1	$\ldots$	1	1	1
t_1	0	0	1	1	$\ldots$	1	1	1
p_1	0	0	0	1	$\ldots$	1	1	1
t_2	0	0	0	0	$\ldots$	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_{k-1}	0	0	0	0	$\ldots$	0	1	1
t_k	0	0	0	0	$\ldots$	0	0	1
p_o	0	0	0	0	$\ldots$	0	0	0

Fig. 14: A workflow net $W_{fin,k}^{\mathrm{CC}}$ with k transitions and $k+1$ places. Its complexity is $C_{\mathrm{CC}}(W_{fin,k}^{\mathrm{CC}}) = 1 - \frac{k \cdot (2k+1)}{(2k+1) \cdot 2k} = \frac{1}{2}$ regardless of the choice of k.
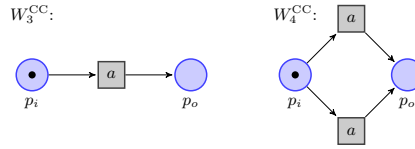
W_3^{CC}: W_4^{CC}:

Fig. 15: Two workflow nets, W_3^{CC} and W_4^{CC} with $L(W_3^{\mathrm{CC}}) = L(W_4^{\mathrm{CC}}) = \{\varepsilon, a\}$, but $C_{\mathrm{CC}}(W_3^{\mathrm{CC}}) = 0$ and $C_{\mathrm{CC}}(W_4^{\mathrm{CC}}) = 1 - \frac{4 \cdot 0.5 + 0.25}{12} = \frac{13}{16}$.

PERM (✓): The workflow nets W_7^{CC} and W_8^{CC} in Figure 17 are permutations of each other but $C_{\mathrm{CC}}(W_7^{\mathrm{CC}}) = 0 \neq \frac{53}{63} = C_{\mathrm{CC}}(W_8^{\mathrm{CC}})$.

ROB$_\ell$ (✓): The cross-connectivity metric is independent of the transition labels, so uniformly changing them has no effect on the complexity score.

¬SUB (✓): Consider the workflow net $W_{\min,1}^{\mathrm{CC}}$ of Figure 18. We get:

- $C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}} \to W_{\min,1}^{\mathrm{CC}}) = 1 - \frac{98}{13 \cdot 12} = \frac{27}{78} \approx 0.3718$
  $> 0.\overline{3} = \frac{1}{6} + \frac{1}{6} = C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}}) + C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}})$,

- $C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}} \wedge W_{\min,1}^{\mathrm{CC}}) = 1 - \frac{104}{16 \cdot 15} = \frac{17}{30} = 0.5\overline{6}$
  $> 0.\overline{3} = \frac{1}{6} + \frac{1}{6} = C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}}) + C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}})$,

- $C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}} \times W_{\min,1}^{\mathrm{CC}}) = 1 - \frac{92 \cdot 24}{18 \cdot 17} = \frac{95}{136} \approx 0.6985$
  $> 0.\overline{3} = \frac{1}{6} + \frac{1}{6} = C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}}) + C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}})$,

- $C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}} \circlearrowright W_{\min,1}^{\mathrm{CC}}) = 1 - \frac{157 \cdot \overline{407}}{22 \cdot 21} \approx 0.6593$
  $> 0.\overline{3} = \frac{1}{6} + \frac{1}{6} = C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}}) + C_{\mathrm{CC}}(W_{\min,1}^{\mathrm{CC}})$.

DEF (✓): The cross-connectivity metric is always defined, since all workflow nets have at least 2 places and 1 transition and we therefore never divide by 0. Also, we have $C_{\mathrm{CC}}(N) \geq 0$ for all workflow nets $M \in \mathcal{M}$, as shown by the following Theorem:
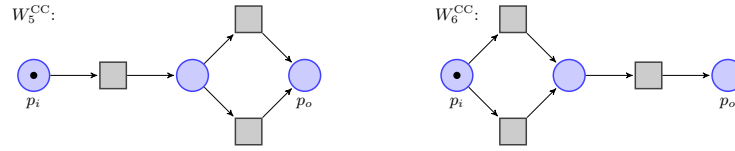
Fig. 16: Two workflow nets, W_5^{CC} and W_6^{CC}, with the same complexity according to the measure C_{CC}.

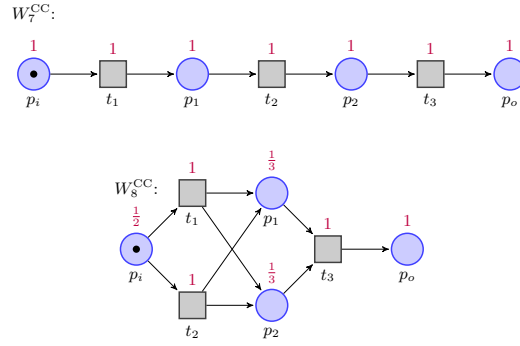Fig. 17: Two workflow nets, W_7^{CC} and W_8^{CC} with $W_7^{CC} \in Perm(W_8^{CC})$, but $C_{CC}(W_7^{CC}) = 0$ and $C_{CC}(W_8^{CC}) = 1 - \frac{2 \cdot 1/2 + 12 \cdot 1/3 + 4 \cdot 1/6 + 1}{42} = \frac{53}{63}$.

Theorem 3. *Let $M = (P, T, F, p_i, p_o) \in \mathcal{M}$ be a workflow net. Then,*

$$\sum_{v_1, v_2 \in P \cup T} V_M(v_1, v_2) \leq (|P| + |T|) \cdot (|P| + |T| - 1).$$

Proof. By definition of V_M, for any nodes $v_1, v_2 \in P \cup T$, $V_M(v_1, v_2) \leq 1$. Therefore, if $V_M(v_1, v_2) = 1$ would be true for all $v_1, v_2 \in P \cup T$, we get

$$\sum_{v_1, v_2 \in P \cup T} V_M(v_1, v_2) \leq (|P| + |T|) \cdot (|P| + |T|).$$

However, it is not possible for all V_M-values to be 1: For any node $v \in P \cup T$, there is no path from v to p_i, since $|^\bullet p_i| = 0$, we get $V_M(v, p_i) = 0$. Furthermore, for any node $v \in P \cup T$ there is no path from p_o to v, since $|p_o^\bullet| = 0$, so $V_M(p_o, v) = 0$. Therefore, there are at most $(|P| + |T| - 1)^2$ nodes v_1, v_2 with $V_N(v_1, v_2) = 1$, giving us

$$\sum_{v_1, v_2 \in P \cup T} V_M(v_1, v_2) \leq (|P| + |T| - 1)^2 \leq (|P| + |T|) \cdot (|P| + |T| - 1)$$
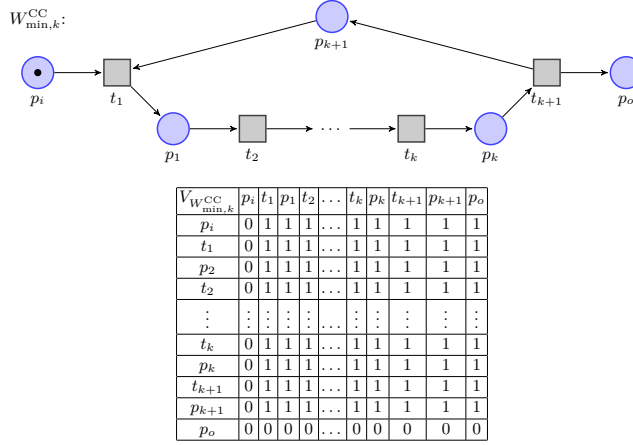
as claimed by the theorem. $\square$

$W^{CC}_{min,k}$:

$V_{W^{CC}_{min,k}}$	p_i	t_1	p_1	t_2	...	t_k	p_k	t_{k+1}	p_{k+1}	p_o
p_i	0	1	1	1	...	1	1	1	1	1
t_1	0	1	1	1	...	1	1	1	1	1
p_2	0	1	1	1	...	1	1	1	1	1
t_2	0	1	1	1	...	1	1	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_k	0	1	1	1	...	1	1	1	1	1
p_k	0	1	1	1	...	1	1	1	1	1
t_{k+1}	0	1	1	1	...	1	1	1	1	1
p_{k+1}	0	1	1	1	...	1	1	1	1	1
p_o	0	0	0	0	...	0	0	0	0	0

Fig. 18: A workflow net with $V_{W^{CC}_{min,k}}(v_1, v_2) = 1$ for all $v_1 \in (P \cup T) \setminus \{p_o\}$ and $v_2 \in (P \cup T) \setminus \{p_i\}$. We get $C_{CC}(W^{CC}_{min,k}) = 1 - \frac{(k+3)+(k+1)-1}{(k+3)+(k+1)} = \frac{1}{2k+4}$.

MIN ($\times$): According to Theorem 3, $C_{CC}(M) \geq 0$ for every workflow net $M \in \mathcal{M}$. However, no workflow net can get the complexity score 0, since this would mean that $V_N(v_1, v_2) = 1$ for every pair of nodes v_1, v_2 with $v_1 \neq p_i$. But this can never happen since $V_N(p_o, v) = 0$ for every node v. Therefore, let $c \in \mathbb{R}$ be a real-number with $c > 0$ and M be a workflow net with complexity $C_{CC}(M) = c$. If we choose a natural number $k > \frac{1}{c}$ and construct the workflow net $W_{min,k}$ of Figure 18, we get $C_{CC}(W_{min,k}) = \frac{1}{2k+4} < \frac{1}{1/c} = c$. Therefore, C_{CC} has no minimal value and thus doesn't fulfill this property.

INF ($\checkmark$): Figure 18 shows how to construct infinitely many workflow nets with different complexity, since $\frac{1}{2k+4} > \frac{1}{2(k+1)+4}$ for any $k \in \mathbb{N}$. Therefore, $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{CC}(M) = c\}| \geq |\frac{1}{2k+4} \mid k \in \mathbb{N}\}| = \infty$.

¬SUP ($\checkmark$): The counter-examples for property MON also show that C_{CC} is not superadditive.

ADD ($\times$): The counter-examples for property ¬SUB also show that C_{CC} is not additive.

Token Split The token split complexity measure quantifies the effort to keep track of the markings in a net if it contains concurrent splits. With each concurrent split, the reader must keep track of the created tokens that need to be synchronized later. As an easy estimator for this efford, Mendling [3, p.128] defines this measure as the sum of additional tokens created with each concurrent split. We can directly translate this measure to workflow nets: Let $W = (P, T, F, p_i, p_o)$ be a workflow net.

$$C_{ts}(W) = \sum_{t \in T}(|t^\bullet| - 1) \tag{9}$$

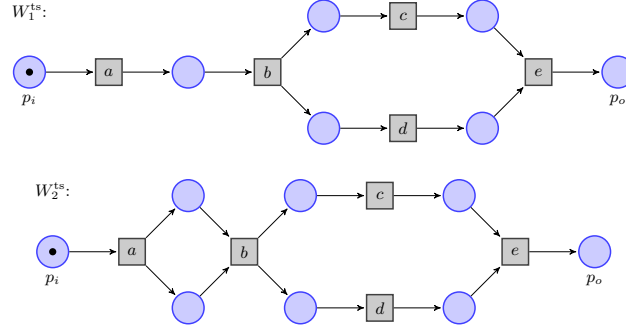Figure 19 shows two example nets and their respective complexity scores.

Fig. 19: Two workflow nets, W_1^{ts} and W_2^{ts}, with different complexity scores according to the token split metric: $C_{\text{ts}}(W_1^{\text{ts}}) = 1$ and $C_{\text{ts}}(W_2^{\text{ts}}) = 2$.

$\neg$**TRIV** ($\checkmark$): For the two nets W_1^{ts} and W_2^{ts} of Figure 19, we get the complexity scores $C_{\text{ts}}(W_1^{\text{ts}}) = 1 \neq 2 = C_{\text{ts}}(W_2^{\text{ts}})$.

FIN ($\times$): Let $c \in \mathbb{N}$ be an arbitrary integer. Figure 20 shows how to construct infinitely many workflow nets with complexity c.
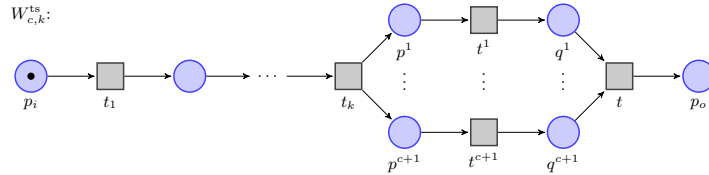
Fig. 20: A construction plan for workflow nets $W_{c,k}^{\text{ts}}$ with $k + c + 2$ transitions and $C_{\text{ts}}(W_{c,k}^{\text{ts}}) = c$ for any $c, k \in \mathbb{N}$.

COL ($\checkmark$): Figure 20 shows how to construct structurally different workflow nets that get the same complexity score according to C_{ts}.

IND$_L$ ($\checkmark$): The two workflow nets W_1^{ts} and W_2^{ts} of Figure 19 have the same language $L(W_1^{\text{ts}}) = \{\varepsilon, a, ab, abc, abd, abcd, abdc, abcde, abdce\} = L(W_2^{\text{ts}})$, but receive complexity scores $C_{\text{ts}}(W_1^{\text{ts}}) = 1 \neq 2 = C_{\text{ts}}(W_2^{\text{ts}})$.

MON ($\checkmark$): To see that C_{ts} fulfills this property, we use the following Theorem:

Theorem 4. *Let $n \geq 2$. For any workflow nets $M_1, \ldots, M_n \in \mathcal{M}$ we get:*
- *$C_{ts}(\rightarrow (M_1, \ldots, M_n)) = C_{ts}(M_1) + \cdots + C_{ts}(M_n)$,*
- *$C_{ts}(\wedge (M_1, \ldots, M_n)) = C_{ts}(M_1) + \cdots + C_{ts}(M_n) + n - 1$,*

- $C_{ts}(\times(M_1, \ldots, M_n)) = C_{ts}(M_1) + \cdots + C_{ts}(M_n)$ *and*
- $C_{ts}(\circlearrowleft(M_1, \ldots, M_n)) = C_{ts}(M_1) + \cdots + C_{ts}(M_n)$.

Proof. The claim is obivous for the operations $\to$, $\times$ and $\circlearrowleft$, since none of the out-going arrows of transitions in $M_1, \ldots, M_n$ are changed and no new transition with more than one out-going arc is introduced by these operations. For $\wedge$, however, we introduce the new transition t_i^* that has n out-going arcs. Therefore, the token split complexity increases by $n-1$ in this case. $\qquad\square$

Thus, according to Theorem 4, $C_{\text{ts}}(M_1 \oplus M_2) \geq C_{\text{ts}}(M_1)$ for any workflow nets $M_1, M_2 \in \mathcal{M}$ and any operation $\oplus \in \{\to, \wedge, \times, \circlearrowleft\}$.

COMP ($\times$): Let $M_1, M_2, M_3 \in \mathcal{M}$ be three arbitrary workflow nets with $C_{\text{ts}}(M_1) = C_{\text{ts}}(M_2)$. Theorem 4 implies:

- $\begin{aligned}[t] C_{\text{ts}}(M_1 \to M_3) &= C_{\text{ts}}(M_1) + C_{\text{ts}}(M_3) \\ &= C_{\text{ts}}(M_2) + C_{\text{ts}}(M_3) = C(M_2 \to M_3), \end{aligned}$
- $\begin{aligned}[t] C_{\text{ts}}(M_1 \wedge M_3) &= C_{\text{ts}}(M_1) + C_{\text{ts}}(M_3) + n - 1 \\ &= C_{\text{ts}}(M_2) + C_{\text{ts}}(M_3) + n - 1 = C(M_2 \wedge M_3), \end{aligned}$
- $\begin{aligned}[t] C_{\text{ts}}(M_1 \times M_3) &= C_{\text{ts}}(M_1) + C_{\text{ts}}(M_3) \\ &= C_{\text{ts}}(M_2) + C_{\text{ts}}(M_3) = C(M_2 \times M_3) \text{ and} \end{aligned}$
- $\begin{aligned}[t] C_{\text{ts}}(M_1 \circlearrowleft M_3) &= C_{\text{ts}}(M_1) + C_{\text{ts}}(M_3) \\ &= C_{\text{ts}}(M_2) + C_{\text{ts}}(M_3) = C(M_2 \circlearrowleft M_3). \end{aligned}$

Therefore, $C_{\text{ts}}(M_1 \oplus M_3) = C_{\text{ts}}(M_2 \oplus M_3)$ for any $\oplus \in \{\to, \wedge, \times, \circlearrowleft\}$.

PERM ($\checkmark$): Figure 21 shows two workflow nets, W_3^{ts} and W_4^{ts} that are permutations of each other, but $C_{\text{ts}}(W_4^{\text{ts}}) = 3 \neq 2 = C_{\text{ts}}(W_3^{\text{ts}})$.
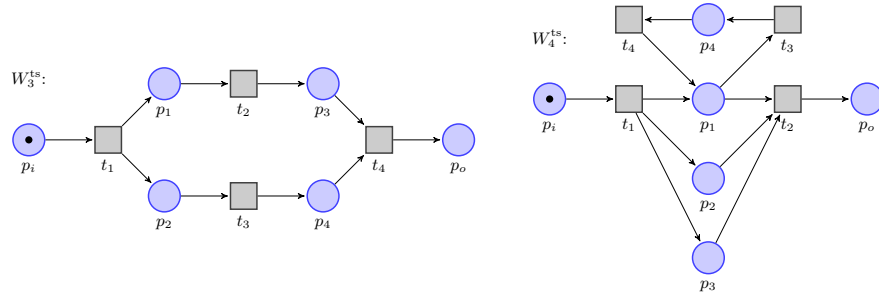
Fig. 21: Two workflow nets, W_3^{ts} and W_4^{ts}, with $W_4^{\text{ts}} \in Perm(W_3^{\text{ts}})$, $C_{\text{ts}}(W_3^{\text{ts}}) = 1$ and $C_{\text{ts}}(W_4^{\text{ts}}) = 2$.

ROB$_\ell$ ($\checkmark$): Changing the labeling function neither changes the set of transitions nor their postsets, so C_{ts} is robust against relabelings.

¬SUB ($\times$): Theorem 4 directly implies that only the operation $\wedge$ is not subadditive. For all other operations, the number of token splits does not strictly increase, no matter which workflow nets we use as input.

DEF ($\checkmark$): The set of transitions T is defined for all workflow nets, as well as the postset of a transition. Furthermore, C_{ts} only returns non-negative values, since every summand is non-negative: If there was a negative summand in C_{ts}, this would mean that there is a transition $t \in T$ with $t^{\bullet} = \emptyset$. But this is impossible in a workflow net, since t must lie on a path from p_i to p_o and therefore needs at least one outgoing edge.

MIN ($\checkmark$): For the smallest possible workflow net, W_0, we get $C_{\text{ts}}(W_0) = 0$. We know from the analysis of property DEF that $C_{\text{ts}}(M) \geq 0$ for all workflow nets $M \in \mathcal{M}$, so C_{ts} reaches its minimum value for the workflow net W_0.

INF ($\checkmark$): Figure 20 shows the construction of workflow nets $W_{c,k}^{\text{ts}}$ with complexity score c for any $c, k \in \mathbb{N}$. Therefore, we can construct infinitely many nets with complexity score c: $|\{c \in \mathbb{N} \mid \exists M \in \mathcal{M} : C_{\text{ts}}(M) = c\}| \geq |\mathbb{N}| = \infty$.

¬SUP ($\checkmark$): Theorem 4 directly implies that C_{ts} is not superadditive for any of the operations $\rightarrow, \wedge, \times, \circlearrowleft$.

ADD ($\times$): Theorem 4 shows that C_{ts} fulfills this property for the operations $\rightarrow, \times$ and $\circlearrowleft$, but not for the operation $\wedge$. Thus, the property holds not for all of the operations of Definition 6.

Separability Separability measures how easy it is to analyze parts of a model in isolation. The idea is that a process model is easier to understand if we can break it down into multiple isolated pieces. Mendling [3, p.122] proposes a separability metric based on cut-vertices in the graph of an EPC model. In graph theory, cut-vertices are vertices in a graph whose removal would split the graph into at least two connected components. For an undirected graph $G = (V, E)$, the set of its connected components is

$$Conn(G) := \{G' = (V', E') \mid \forall u, v \in V' : \text{there is a path from } u \text{ to } v \text{ in } G'\}.$$

With this definition, the set of cut-vertices of an undirected graph G is

$$\mathcal{V}(G) := \{v \in V \mid |Conn((V \setminus \{v\}, E \setminus \{\{x, y\} \mid x = v \vee y = v\}))| \geq 2\}.$$

This definition can also be applied to directed graphs by ignoring the direction of arcs in the graph.

Mendling defines the separability measure as the number of cut-vertices of the process model divided by the total number of vertices. We can directly translate this to workflow nets, but keep in mind that the input- and output places of a workflow net can never be cut-vertices. Let $W = (P, T, F, p_i, p_o)$ be a workflow net and $\overline{W} = (P \cup T, \{\{x, y\} \mid (x, y) \in F \vee (y, x) \in F\})$ be the undirected graph produced by ignoring the arc directions in W. The separability measure is then defined as

$$C_{\text{sep}}(W) = 1 - \frac{|\mathcal{V}(\overline{W})|}{|P| + |T| - 2} \tag{10}$$

In contrast to the original definition, we subtract the fraction from 1 to ensure that workflow nets with many cut-vertices get a low complexity score. This way, a low complexity score refers to less complex models. Figure 22 shows three example nets and their respective complexity scores.
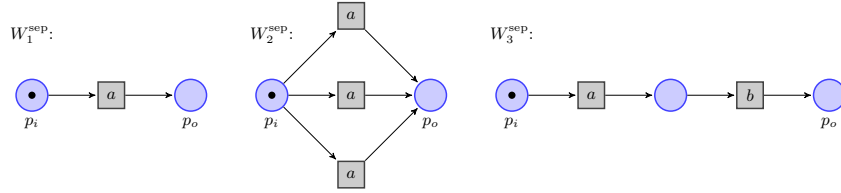
Fig. 22: Three workflow nets, W_1^{sep}, W_2^{sep}, W_3^{sep}, with $C_{\text{sep}}(W_2^{\text{sep}}) = 1$ and $C_{\text{sep}}(W_1^{\text{sep}}) = C_{\text{sep}}(W_3^{\text{sep}}) = 0$.

¬**TRIV** (✓): For the two nets W_1^{sep} and W_2^{sep} of Figure 22, we get the complexity scores $C_{\text{sep}}(W_1^{\text{sep}}) = 0 \neq 1 = C_{\text{sep}}(W_2^{\text{sep}})$.

FIN (×): For any $n \in \mathbb{N}$, the workflow net $W_{fin,n}^{\text{sep}}$ of Figure 23, we have $C_{\text{sep}}(W_{fin,n}^{\text{sep}}) = 0$. Therefore, there are infinitely many workflow nets with this complexity score.
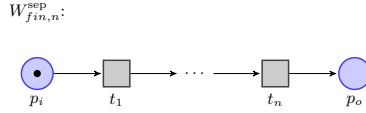
Fig. 23: A worklfow net with complexity 0 and n transitions.

COL (✓): The workflow nets W_1^{sep} and W_3^{sep} of Figure 22 are different in structure, but both get the same complexity score $C_{\text{sep}}(W_1^{\text{sep}}) = 0 = C_{\text{sep}}(W_3^{\text{sep}})$.

IND$_L$ (✓): Take the workflow nets W_1^{sep} and W_2^{sep} of Figure 22. Their languages are $L(W_1^{\text{sep}}) = \{\varepsilon, a\} = L(W_2^{\text{sep}})$, but their complexity scores are $C_{\text{sep}}(W_1^{\text{sep}}) = 0 \neq 1 = C_{\text{sep}}(W_2^{\text{sep}})$.

MON (×): The workflow net W_2^{sep} of Figure 22 contains no cut-vertices, so $C_{\text{sep}}(W_2^{\text{sep}}) = 1$. But, since all operations of Definition 6 except × introduce at least one cut-vertex, we get:

- $C_{\text{sep}}(W_2^{\text{sep}} \to W_2^{\text{sep}}) = 1 - \frac{3}{9} = \frac{2}{3} < 1 = C_{\text{sep}}(W_2^{\text{sep}})$,
- $C_{\text{sep}}(W_2^{\text{sep}} \wedge W_2^{\text{sep}}) = 1 - \frac{2}{12} = \frac{5}{6} < 1 = C_{\text{sep}}(W_2^{\text{sep}})$ and
- $C_{\text{sep}}(W_2^{\text{sep}} \circlearrowleft W_2^{\text{sep}}) = 1 - \frac{2}{20} = \frac{9}{10} < 1 = C_{\text{sep}}(W_2^{\text{sep}})$.

For the operator ×, Figure 24 shows two workflow nets, W_5^{sep} and W_6^{sep}, where $C_{\text{sep}}(W_6^{\text{sep}} \times W_5^{\text{sep}}) \approx 0.9286 < 1 = C_{\text{sep}}(W_6^{\text{sep}})$.

COMP (✓): Figure 25 shows workflow nets W_7^{sep} and W_8^{sep} with the same complexity score $C_{\text{sep}}(W_7^{\text{sep}}) = 0.25 = C_{\text{sep}}(W_8^{\text{sep}})$. But if we combine these nets with W_0 of Figure 2, we get:

- $C_{\text{sep}}(W_7^{\text{sep}} \to W_0) = 1 - \frac{7}{8} = \frac{1}{8} \neq \frac{1}{6} = 1 - \frac{10}{12} = C_{\text{sep}}(W_8^{\text{sep}} \to W_0)$,
- $C_{\text{sep}}(W_7^{\text{sep}} \wedge W_0) = 1 - \frac{3}{11} = \frac{8}{11} \neq \frac{13}{15} = 1 - \frac{2}{15} = C_{\text{sep}}(W_8^{\text{sep}} \wedge W_0)$,
- $C_{\text{sep}}(W_7^{\text{sep}} \times W_0) = 1 - \frac{1}{13} = \frac{12}{13} \neq 1 = 1 - \frac{0}{17} = C_{\text{sep}}(W_8^{\text{sep}} \times W_0)$,
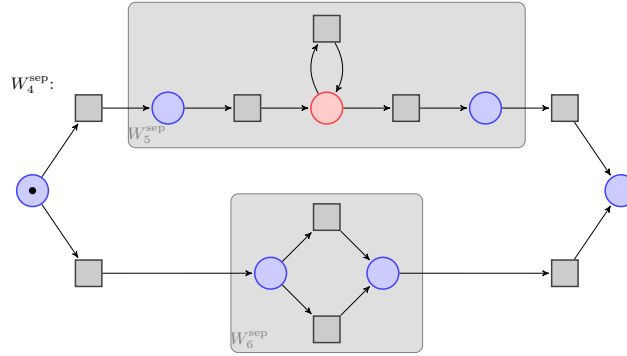
Fig. 24: A workflow net W_4^{sep} with $W_4^{\text{sep}} = \times(W_5^{\text{sep}}, W_6^{\text{sep}})$. Computing the separability metric yields $C_{\text{sep}}(W_4^{\text{sep}}) = 1 - \frac{1}{14} \approx 0.9286$, $C_{\text{sep}}(W_5^{\text{sep}}) = 1 - \frac{1}{4} = 0.75$ and $C_{\text{sep}}(W_6^{\text{sep}}) = 1 - \frac{0}{2} = 1$.
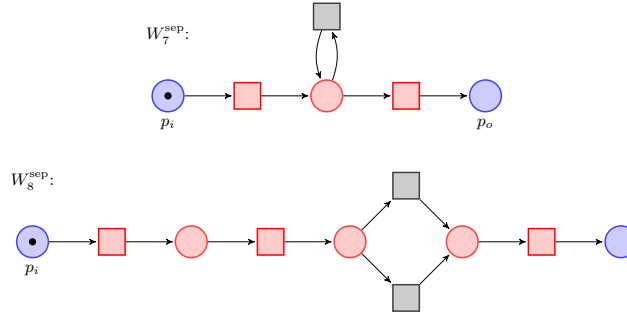

Fig. 25: Two workflow nets, W_7^{sep}, W_8^{sep}, with $C_{\text{sep}}(W_7^{\text{sep}}) = C_{\text{sep}}(W_8^{\text{sep}}) = 0.25$.

– $C_{\text{sep}}(W_7^{\text{sep}} \circlearrowleft W_0) = 1 - \frac{5}{17} = \frac{12}{17} \neq \frac{17}{21} = 1 - \frac{4}{21} = C_{\text{sep}}(W_8^{\text{sep}} \circlearrowleft W_0)$.

PERM ($\checkmark$): Figure 26 shows two workflow nets, W_9^{sep} and W_{10}^{sep} that are permutations of each other, but $C_{\text{sep}}(W_9^{\text{sep}}) = 0.25 \neq 0.5 = C_{\text{sep}}(W_{10}^{\text{sep}})$.

ROB$_\ell$ ($\checkmark$): The number of cut-vertices is independent of the labeling. Uniformly changing the labeling therefore has no impact on the complexity score of C_{sep}.

¬SUB ($\times$): C_{sep} fulfills this property only for the operations $\wedge$, $\times$ and $\circlearrowleft$: Take the workflow net W_0 of Figure 2. We get:

– $C_{\text{sep}}(W_0 \wedge W_0) = 1 - \frac{2}{8} = 0.75 > 0 + 0 = C_{\text{sep}}(W_0) + C_{\text{sep}}(W_0)$,
– $C_{\text{sep}}(W_0 \times W_0) = 1 - \frac{0}{10} = 1 > 0 + 0 = C_{\text{sep}}(W_0) + C_{\text{sep}}(W_0)$,
– $C_{\text{sep}}(W_0 \circlearrowleft W_0) = 1 - \frac{4}{14} \approx 0.7143 > 0 + 0 = C_{\text{sep}}(W_0) + C_{\text{sep}}(W_0)$.

For the operation $\rightarrow$, however, C_{sep} is subadditive. We show this by first proving the following Theorem

Theorem 5. *Let $M_1, M_2 \in \mathcal{M}$ be two workflow nets. Further, let $p_o^{M_1}$ be the output place of M_1, $p_i^{M_2}$ be the input place of M_2 and $t_\rightarrow$ be the newly*
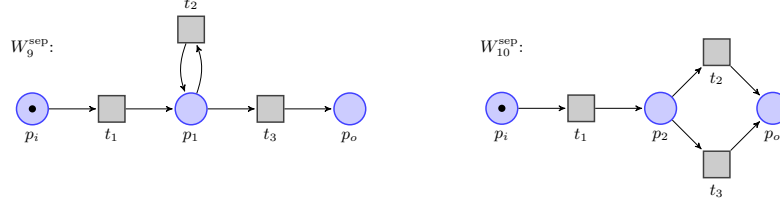
Fig. 26: Two workflow nets with $W_9^{\text{sep}} \in Perm(W_{10}^{\text{sep}})$ and complexity scores $C_{\text{sep}}(W_9^{\text{sep}}) = 0.25 \neq 0.5 = C_{\text{sep}}(W_{10}^{\text{sep}})$.

introduced transition connecting $p_o^{M_1}$ and $p_i^{M_2}$. Then, the set of cut-vertices in the net $M_1 \to M_2$ is $\mathcal{V}(\overline{M_1 \to M_2}) = \mathcal{V}(\overline{M_1}) \cup \mathcal{V}(\overline{M_2}) \cup \{p_o^{M_1}, p_i^{M_2}, t_\to\}$.

Proof. It is obvious by construction that $p_o^{M_1}$, $p_i^{M_2}$ and $t_\to$ are cut-vertices in $M_1 \to M_2$. Therefore, let v be a cut-vertex in M_1 or M_2. Without loss of generality, let v be a cut-vertex of M_1. Then, removing v leads to more than one connected component in M_1. Because the $\to$-operation only introduces the edges $(p_o^{M_1}, t_\to)$ and $(t_\to p_i^{M_2})$, removing v in $M_1 \to M_2$ also leads to more than one connected component. Therefore, v is also a cut-vertex in $M_1 \to M_2$.

Wlog let $\overline{v}$ be a vertex in M_1 that is not a cut-vertex. Then, for each pair of vertices, there is a path connecting them that avoids $\overline{v}$. Since we don't remove any arcs with the $\to$-operation, this path still exists in $M_1 \to M_2$, so $\overline{v}$ is also not a cut-vertex in this net. $\qquad\square$

Because of Theorem 5, we know that for any nets $M_1, M_2 \in \mathcal{M}$, the set of non-cut-vertices in $M_1 \to M_2$ is the union of the set of non-cut-vertices in M_1 and M_2. Let $M_1, M_2 \in \mathcal{M}$, x_1 be the number of non-cut-vertices in M_1, x_2 the number of non-cut-vertices in M_2 and n_1, n_2 be the number of nodes in M_1, M_2 respectively. We then get:

$$
\begin{aligned}
C_{\text{sep}}(M_1) + C_{\text{sep}}(M_2) &= \frac{x_1}{n_1 - 2} + \frac{x_2}{n_2 - 2} > \frac{x_1}{n_1 + n_2 - 2} + \frac{x_2}{n_1 + n_2 - 2} \\
&= \frac{x_1 + x_2}{n_1 + n_2 - 2} = C_{\text{sep}}(M_1 \to M_2)
\end{aligned}
$$

because $n_1, n_2 \geq 3$. Therefore, C_{sep} fulfills this property for $\to$.

DEF ($\checkmark$): The separability measure is defined for all workflow nets, since the set of cut-vertices is defined for all graphs. The metric returns only non-negative values, since the input and output place can never be cut-vertices, so we get $|\mathcal{V}(\overline{W})| \leq |P| + |T| - 2$.

MIN ($\checkmark$): As shown for property DEF, C_{sep} returns no value below 0. In the workflow net W_0 of Figure 2, every vertex except for the input and output place are cut-vertices, so $C_{\text{sep}}(W_0) = 0$ is the minimum complexity score.

INF ($\checkmark$): Figure 27 shows how to construct workflow nets with $3n + 2m + 2$ nodes for any $n, m \in \mathbb{N}$, such that $C_{\text{sep}}(W_{n,m}^{\text{sep}}) = 1 - \frac{n+2}{3n+2m}$. Therefore, $|\{c \mid \exists M \in \mathcal{M} : C_{\text{sep}}(M) = c\}| \geq |\{r \in \mathbb{R} \mid \exists n, m \in \mathbb{N} : r = 1 - \frac{n+2}{3n+2m}\}| = \infty$, so C_{sep} fulfills this property.
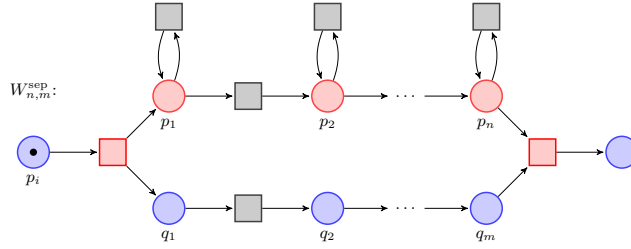

Fig. 27: A worklfow net with $3n + 2m + 2$ nodes. All nodes highlighted in red are cut vertices, so its complexity is $C_{\text{sep}}(W_{n,m}^{\text{sep}}) = 1 - \frac{n+2}{3n+2m}$.

$\neg$**SUP** ($\checkmark$): The counter-examples for property MON show that C_{sep} is not superadditive for any of the operations.

ADD ($\times$): The counter-examples for property MON show that C_{sep} is not additive for any of the operations.

Control Flow Complexity Similar to token-split complexity, the control flow complexity [5] aims to quantify how many different possible control flows the user needs to keep track of to fully understand the net. In contrast to the token split metric, however, control flow complexity considers not only parallel splits, but all connector types of the net. Cardoso [5] argues that parallel splits in BPMN models don't increase the number of possible control flows, while an exclusive choice with k outgoing edges creates k alternative control flows in the net. Since connector nodes are always harder to understand than pure sequences, parallel splits give a small penalty to the complexity score. An inclusive choice is even worse, since it allows for any combination of transitions in the postset, as long as at least one of them fires. For an inclusive choice with k out-going edges, the control flow complexity score increases by $2^k - 1$.
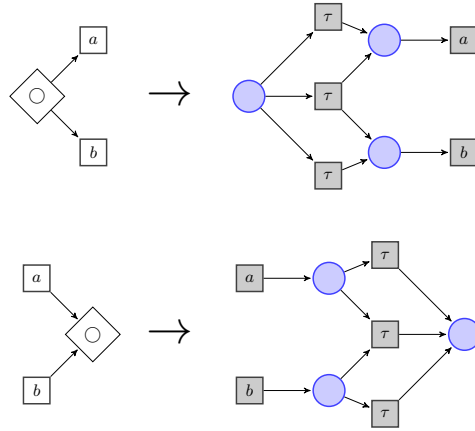

Fig. 28: Inclusive choice connectors in BPMN notation (left) and their translation to workflow nets (right).

In workflow nets, we don't have single nodes that initiate an inclusive choice. Instead, we model such a choice like shown in Figure 28. That way, we can translate any or-split in BPMN notation into such a structure. We need to make sure that this translation would not get a smaller complexity score than an inclusive choice node for BPMN. But this is not the case, since we need to enable all combinations of transitions in the inclusive choice. Therefore, the place labeled p in Figure 28 alone increases the complexity score by $\sum_{i=1}^{k} \binom{k}{i} = 2^k - 1$ if k transitions form an inclusive choice. Apart from p, we have also $2^k - 1 - k$ transitions in the or-split-structure that are parallel splits, as well as k xor-joins in the or-join-structure, further increasing the complexity score. For simplicity, we accept that the penalty for inclusive choices in workflow nets is higher than

in BPMN and define:

$$C_{\text{CFC}}(W) = |\mathcal{S}_{\text{and}}^W| + \sum_{p \in \mathcal{S}_{\text{xor}}^W} |p^\bullet|. \tag{11}$$

Cardoso [5] already analyzed which of Weyuker's properties the control flow complexity measure fulfills. Since we translated the definition to workflow nets, we revisit these analyses and check that they didn't change. Figure 29 shows three example nets and their respective complexity scores.
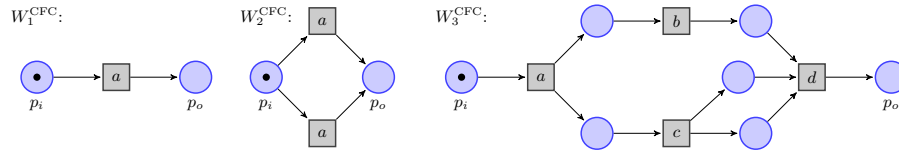

Fig. 29: Three workflow nets, $W_1^{\text{CFC}}, W_2^{\text{CFC}}, W_3^{\text{CFC}}$, with $C_{\text{CFC}}(W_1^{\text{CFC}}) = 0$ and $C_{\text{CFC}}(W_2^{\text{CFC}}) = C_{\text{CFC}}(W_3^{\text{CFC}}) = 2$.

¬**TRIV** ($\checkmark$ [5]): For the two nets W_1^{CFC} and W_2^{CFC} of Figure 29, we get the complexity scores $C_{\text{CFC}}(W_1^{\text{CFC}}) = 0 \neq 2 = C_{\text{CFC}}(W_2^{\text{CFC}})$.

FIN ($\times$ [5]): Let $c \in \mathbb{N}_0$ be an arbitrary integer. Figure 30 shows how to construct infinitely many workflow nets with complexity score c according to the complexity measure C_{CFC}.
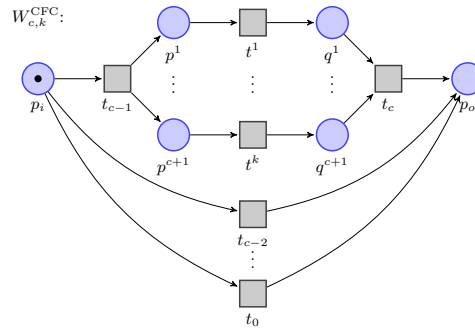

Fig. 30: A construction plan for workflow nets $W_{c,k}^{\text{CFC}}$ with $k + c$ transitions and $C_{\text{CFC}}(W_{c,k}^{\text{CFC}}) = c$ for any $c, k \in \mathbb{N}_0$.

COL ($\checkmark$ [5]): The workflow nets W_2^{CFC} and W_3^{CFC} of Figure 29 are different in structure, but both have complexity $C_{\text{CFC}}(W_2^{\text{CFC}}) = 2 = C_{\text{CFC}}(W_3^{\text{CFC}})$.

IND$_L$ ($\checkmark$ [5]): The two workflow nets W_1^{CFC} and W_2^{CFC} of Figure 29 have the same language $L(W_1^{\mathrm{CFC}}) = \{\varepsilon, a\} = L(W_2^{\mathrm{CFC}})$, but receive complexity scores $C_{\mathrm{CFC}}(W_1^{\mathrm{CFC}}) = 0 \neq 2 = C_{\mathrm{CFC}}(W_2^{\mathrm{CFC}})$.

MON ($\checkmark$ [5]): To show that C_{CFC} is monotone, we prove the following:

> **Theorem 6.** *Let $n \geq 2$. For any workflow nets $M_1, \ldots, M_n \in \mathcal{M}$ we get:*
> - $C_{CFC}(\rightarrow (M_1, \ldots, M_n)) = C_{CFC}(M_1) + \cdots + C_{CFC}(M_n)$,
> - $C_{CFC}(\wedge(M_1, \ldots, M_n)) = C_{CFC}(M_1) + \cdots + C_{CFC}(M_n) + 1$,
> - $C_{CFC}(\times(M_1, \ldots, M_n)) = C_{CFC}(M_1) + \cdots + C_{CFC}(M_n) + n$ *and*
> - $C_{CFC}(\circlearrowleft (M_1, \ldots, M_n)) = C_{CFC}(M_1) + \cdots + C_{CFC}(M_n) + n$.
>
> *Proof.* None of the operations changes the pre- or postset of existing places or transitions in $M_1, \ldots, M_n$, except for the input and output place. Furthermore, the operation $\rightarrow$ doesn't introduce new and- or xor-splits, so $C_{\mathrm{CFC}}(\rightarrow (M_1, \ldots, M_n)) = C_{\mathrm{CFC}}(M_1) + \cdots + C_{\mathrm{CFC}}(M_n)$. The operation $\wedge$ adds one and-split, which increases the complexity by 1. Operations $\times$ and $\circlearrowleft$ each introduce one new xor-split that has n outgoing edges, thus increasing the complexity score by n. $\qquad\square$

Therefore, according to Theorem 6, $C_{\mathrm{CFC}}(M_1 \oplus M_2) \geq C_{\mathrm{CFC}}(M_1)$ for any workflow nets $M_1, M_2 \in \mathcal{M}$ and any operation $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$.

COMP ($\times$ [5]): Let $M_1, M_2, M_3 \in \mathcal{M}$ with $C_{\mathrm{CFC}}(M_1) = C_{\mathrm{CFC}}(M_2)$. Theorem 6 gives us the following equalities:
- $C_{\mathrm{CFC}}(M_1 \rightarrow M_3) = C_{\mathrm{CFC}}(M_1) + C_{\mathrm{CFC}}(M_3)$
  $\qquad\qquad = C_{\mathrm{CFC}}(M_2) + C_{\mathrm{CFC}}(M_3) = C_{\mathrm{CFC}}(M_1 \rightarrow M_3)$,
- $C_{\mathrm{CFC}}(M_1 \wedge M_3) = C_{\mathrm{CFC}}(M_1) + C_{\mathrm{CFC}}(M_3) + 1$
  $\qquad\qquad = C_{\mathrm{CFC}}(M_2) + C_{\mathrm{CFC}}(M_3) + 1 = C_{\mathrm{CFC}}(M_1 \wedge M_3)$,
- $C_{\mathrm{CFC}}(M_1 \times M_3) = C_{\mathrm{CFC}}(M_1) + C_{\mathrm{CFC}}(M_3) + n$
  $\qquad\qquad = C_{\mathrm{CFC}}(M_2) + C_{\mathrm{CFC}}(M_3) + n = C_{\mathrm{CFC}}(M_1 \times M_3)$,
- $C_{\mathrm{CFC}}(M_1 \circlearrowleft M_3) = C_{\mathrm{CFC}}(M_1) + C_{\mathrm{CFC}}(M_3) + n$
  $\qquad\qquad = C_{\mathrm{CFC}}(M_2) + C_{\mathrm{CFC}}(M_3) + n = C_{\mathrm{CFC}}(M_1 \circlearrowleft M_3)$.

PERM ($\checkmark$ [5]): Figure 31 shows two workflow nets, W_4^{CFC} and W_5^{CFC}, that are permutations of each other, but $C_{\mathrm{CFC}}(W_4^{\mathrm{CFC}}) = 2 \neq 4 = C_{\mathrm{CFC}}(W_5^{\mathrm{CFC}})$.

ROB$_\ell$ ($\checkmark$ [5]): A uniform relabelling of the transitions has no effect on the set of split nodes or their out-degree. Therefore, uniformly changing the transition labels has no impact on the complexity score assigned by C_{CFC}.

$\neg$SUB ($\times$ [5]): Theorem 6 implies $C_{\mathrm{CFC}}(M_1 \oplus M_2) > C_{\mathrm{CFC}}(M_1) + C_{\mathrm{CFC}}(M_2)$ for all $M_1, M_2 \in \mathcal{M}$ and $\oplus \in \{\wedge, \times, \circlearrowleft\}$. For the operation $\rightarrow$ however, C_{CFC} fulfills only a weaker property: $C_{\mathrm{CFC}}(M_1 \rightarrow M_2) \geq C_{\mathrm{CFC}}(M_1) + C_{\mathrm{CFC}}(M_2)$ for all $M_1, M_2 \in \mathcal{M}$.

DEF ($\checkmark$): S_{and}^W and S_{xor}^W are defined for each workflow net, so C_{CFC} is also defined for each workflow net. Furthermore, S_{and}^W, S_{xor}^W and $p^\bullet$ for any $p \in P$ are sets, so their cardinality can't get below 0. Thus, $C_{\mathrm{CFC}}(W) \geq 0$ for any workflow net W.

MIN ($\checkmark$): The smallest possible value C_{CFC} can return is 0. The workflow net W_0 of Figure 2 is one example of a workflow net that gets this complexity score, since it doesn't have any connector nodes.
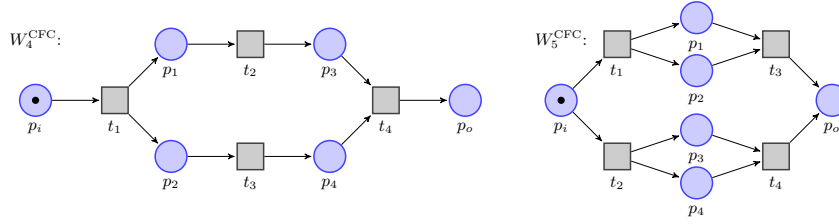
Fig. 31: Two workflow nets, W_4^{CFC} and W_5^{CFC}, with $W_5^{\mathrm{CFC}} \in Perm(W_4^{\mathrm{CFC}})$, $C_{\mathrm{CFC}}(W_4^{\mathrm{CFC}}) = 2$ and $C_{\mathrm{CFC}}(W_5^{\mathrm{CFC}}) = 4$.

INF ($\checkmark$): Let $c \in \mathbb{N}_0$. Figure 30 shows how to construct workflow nets with complexity c. Therefore, $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{CFC}}(M) = c\}| \geq |\mathbb{N}_0| = \infty$.

¬SUP ($\times$): Theorem 6 implies that C_{CFC} does not fulfill this property for any of the operations $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowright\}$.

ADD ($\times$): Theorem 6 implies that C_{CFC} fulfills this property only for the operation $\rightarrow$, but not for the other operations of Definition 6.

5.2   Node IO Complexity

The second complexity dimension found by Lieben et al. [9] is labeled Node IO Complexity. Measures of this dimension take the number of incoming and outgoing sequence flows into account. Apart from the complexity measures we discuss in this section, the Coefficient of Network Connectivity [3, p.120] also belongs to this dimension. This measure has a special role, as it belongs to two of the four dimensions. We therefore analyze this measure within the section of the fourth dimension to keep the dimensions equally large.

Maximum Connector Degree The maximum connector degree is the maximum number of arcs that enter or leave a single connector node in the net [3, p121]. Connectors with a high degree introduce complexity, since they either increase the number of possible control flows or the amount of tokens to keep track of.

$$C_{\mathrm{mcd}}(W) = \max\{\deg(x) \mid x \in \mathcal{C}^W\} \tag{12}$$

Figure 32 shows three example nets and their respective complexity scores.

¬TRIV ($\checkmark$): For the two nets W_1^{mcd} and W_2^{mcd} of Figure 32, we get the complexity scores $C_{\mathrm{mcd}}(W_1^{\mathrm{mcd}}) = 2 \neq 3 = C_{\mathrm{mcd}}(W_2^{\mathrm{mcd}})$.

FIN ($\times$): Follows directly by Theorem 1. Figure 33 shows how to construct such a sequence of places and transitions.

COL ($\checkmark$): The workflow nets W_2^{mcd} and W_3^{mcd} of Figure 32 are different in structure, but have equal complexity: $C_{\mathrm{mcd}}(W_2^{\mathrm{mcd}}) = 3 = C_{\mathrm{mcd}}(W_3^{\mathrm{mcd}})$.
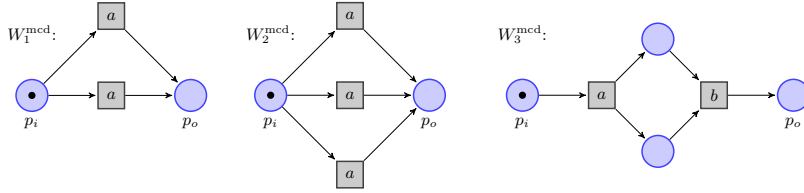
Fig. 32: Example nets W_1^{mcd}, W_2^{mcd} and W_3^{mcd}, with $C_{\mathrm{mcd}}(W_1^{\mathrm{mcd}}) = 2$ and $C_{\mathrm{mcd}}(W_2^{\mathrm{mcd}}) = C_{\mathrm{mcd}}(W_3^{\mathrm{mcd}}) = 3$.
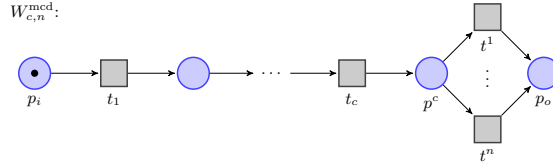

Fig. 33: A workflow net $W_{c,n}^{\mathrm{mcd}}$ with $c+n$ transitions, $c+2$ places and complexity $C_{\mathrm{mcd}}(W_{c,n}^{\mathrm{mcd}}) = n+1$.

IND$_L$ ($\checkmark$): The workflow nets W_1^{mcd} and W_2^{mcd} of Figure 32 have the same language $L(W_1^{\mathrm{mcd}}) = \{\varepsilon, a\} = L(W_2^{\mathrm{mcd}})$, but receive different complexity scores $C_{\mathrm{mcd}}(W_1^{\mathrm{mcd}}) = 2 \neq 3 = C_{\mathrm{mcd}}(W_3^{\mathrm{mcd}})$.

MON ($\checkmark$): Let $M_1, M_2 \in \mathcal{M}$ be workflow nets and let $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$. Then, there are only two cases:

1. $M_1 \oplus M_2$ contains the connector with the highest degree. Then, obviously $C_{\mathrm{mcd}}(M_1 \oplus M_2) \geq C_{\mathrm{mcd}}(M_1)$.
2. M_1 or M_2 contain the connector with the highest degree. Wlog let M_2 be the net with this connector. Let $C_{\mathrm{mcd}}(M_2) = d$ and v be a connector with this degree in M_2. Then, in $M_1 \oplus M_2$, the node v has either degree d or degree $d+1$, if a new arc to v was added with the operation. Therefore, $C_{\mathrm{mcd}}(M_1 \oplus M_2) \geq d \geq C_{\mathrm{mcd}}(M_1)$.

COMP ($\checkmark$): Take the workflow nets W_2^{mcd} and W_3^{mcd} of Figure 32, as well as W_0 of Figure 2. We know that $C_{\mathrm{mcd}}(W_2^{\mathrm{mcd}}) = 3 = C_{\mathrm{mcd}}(W_3^{\mathrm{mcd}})$, but regardless of which operation $\oplus \in \{\rightarrow, \wedge, \circlearrowleft, \times\}$ we use, we get the result $C_{\mathrm{mcd}}(W_2^{\mathrm{mcd}} \oplus W_0) = 4 \neq 3 = C_{\mathrm{mcd}}(W_3^{\mathrm{mcd}} \oplus W_0)$.

PERM ($\checkmark$): Figure 34 shows two workflow nets, W_4^{mcd} and W_5^{mcd}, that are permutations of each other, but $C_{\mathrm{mcd}}(W_4^{\mathrm{mcd}}) = 3 \neq 4 = C_{\mathrm{mcd}}(W_5^{\mathrm{mcd}})$.

ROB$_\ell$ ($\checkmark$): The C_{mcd} measure is robust against relabellings, as can be seen straightforwardly by its insensitivity to transition labels generally.

¬SUB ($\times$): If we consider only input workflow nets where C_{mcd} is defined, and so can assume that all input workflow nets have at least one connector, C_{mcd} is subadditive for the operations $\rightarrow$, $\wedge$, $\times$ and $\circlearrowleft$.
Let $M_1, M_2 \in \mathcal{M}$ be two workflow nets. We know that $C_{\mathrm{mcd}}(M_1) \geq 2$ and $C_{\mathrm{mcd}}(M_2) \geq 2$, since a node only qualifies as a connector if it has at least two
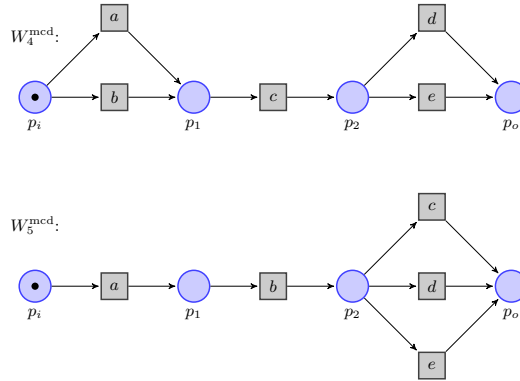
Fig. 34: Two workflow nets, W_4^{mcd} and W_5^{mcd}, with $W_4^{\mathrm{mcd}} \in Perm(W_5^{\mathrm{mcd}})$, but $C_{\mathrm{mcd}}(W_4^{\mathrm{mcd}}) = 3$ and $C_{\mathrm{mcd}}(W_5^{\mathrm{mcd}}) = 4$.

outgoing or incoming arcs. Let $c_1 := C_{\mathrm{mcd}}(M_1)$ and $c_2 := C_{\mathrm{mcd}}(M_2)$. Then, the maximum connector degree in $M_1 \oplus M_2$ for some $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$ can be $\max c_1, c_2 + 1$ if the connector with maximum degree is an initial or final node in its net. If both M_1 and M_2 have complexity 2, we get $C_{\mathrm{mcd}}(M_1 \oplus M_2) = 3 < 4 = C_{\mathrm{mcd}}(M_1) + C_{\mathrm{mcd}}(M_2)$. Otherwise, we get $C_{\mathrm{mcd}}(M_1 \oplus M_2) \leq \max\{c_1, c_2\} + 1 < 2 \cdot \max\{c_1, c_2\} = C_{\mathrm{mcd}}(M_1) + C_{\mathrm{mcd}}(M_2)$. If the metric is redefined to give nets with no connectors a complexity of zero, subadditivity is contradicted for $\oplus \in \{\wedge, \times, \circlearrowleft\}$: For the net W_0 of Figure 2, we get $C_{\mathrm{mcd}}(W_0 \oplus W_0) \geq 2 > 0 = C_{\mathrm{mcd}}(W_0) + C_{\mathrm{mcd}}(W_0)$ for these operations. For the operation $\rightarrow$, subadditivity still holds in this case.

DEF ($\times$): The maximum connector degree is undefined for workflow nets without any connectors. However, we can add a special case where C_{mcd} returns the complexity score 0 if there are no connectors in the net. With this adaptation, C_{mcd} returns only non-negative values, since the degree of a node is always non-negative.

MIN ($\checkmark$): With the adaptation shown for in the analysis of DEF, the smallest workflow net, W_0 of Figure 2, has two places and one transition. None of them are connectors, making the minimum complexity zero. For workflow nets that contain at least one connector node, the maximum connector degree is ≥ 2, since a connector node needs at least two incoming or outgoing arcs.

INF ($\checkmark$): Let $n \in \mathbb{N}$ with $n \geq 2$. Figure 33 shows how to construct a workflow net with complexity score $n + 1$, so C_{mcd} can return infinitely many values. We get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{mcd}}(M) = c\}| \geq \{n \in \mathbb{N} \mid n \geq 3\} = \infty$.

¬SUP ($\checkmark$): The discussion for ¬SUB shows that C_{mcd} is not superadditive.

ADD ($\times$): The discussion for ¬SUB shows C_{mcd} is not additive.

Sequentiality Naturally, sequences of transitions are easier to understand than concurrent or exclusive execution. Therefore, Mendling [3, p123] defines a complexity measure for sequentiality, which is the number of arcs between non-

connector nodes divided by the total number of arcs. The higher the score of this metric, the lower the amount of direct sequences in the net. In our version of this complexity metric, we therefore subtract this fraction from 1 to ensure comparability with the other complexity measures.

$$C_{\text{seq}}(W) = 1 - \frac{|\{(x, y) \in F \mid x, y \notin \mathcal{C}^W\}|}{|F|} \tag{13}$$

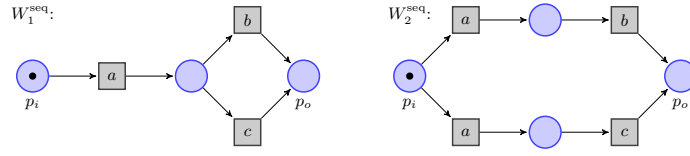Figure 35 shows two example nets and their respective complexity scores.

Fig. 35: Example nets with $C_{\text{seq}}(W_1^{\text{seq}}) = \frac{5}{6}$ and $C_{\text{seq}}(W_2^{\text{seq}}) = \frac{4}{8} = \frac{1}{2}$.

¬**TRIV** (✓): For the two nets W_1^{seq} and W_2^{seq} of Figure 35, we get the complexity scores $C_{\text{seq}}(W_1^{\text{seq}}) = \frac{5}{6} \neq \frac{1}{2} = C_{\text{seq}}(W_2^{\text{seq}})$.

FIN (×): The net $W_{c,k}^{\text{seq}}$ in Figure 36 is a choice between c sequences containing $2k - 1$ arcs. In total, it has $|F| = 2ck + c$ arcs. Of those, $2c$ have at least one connector as an endpoint, so $C_{\text{seq}}(W_{c,k}^{\text{seq}}) = \frac{2c}{2ck+c} = \frac{2}{2k+1}$. As this complexity score is independent of c, we have found infinitely many workflow nets with this complexity score.
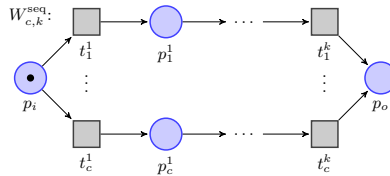

Fig. 36: A workflow net $W_{c,k}^{\text{seq}}$ with complexity $C_{\text{seq}}(W_{c,k}^{\text{seq}}) = \frac{2c}{2ck+c} = \frac{2}{2k+1}$.

COL (✓): The workflow nets $W_{2,3}^{\text{seq}}$ and $W_{3,3}^{\text{seq}}$ of Figure 36 are different in structure, but have equal complexity: $C_{\text{seq}}(W_{2,3}^{\text{seq}}) = \frac{2}{7} = C_{\text{seq}}(W_{3,3}^{\text{seq}})$.

IND$_L$ (✓): The workflow nets W_1^{seq} and W_2^{seq} of Figure 35 have the same language $L(W_1^{\text{seq}}) = \{\varepsilon, a, ab, ac\} = L(W_2^{\text{seq}})$, but receive different complexity scores $C_{\text{seq}}(W_1^{\text{seq}}) = \frac{5}{6} \neq \frac{1}{2} = C_{\text{seq}}(W_2^{\text{seq}})$.

MON ($\times$): Take the workflow nets W_4^{seq} and W_0 of Figure 37. The same Figure already shows that C_{seq} is not monotone for $\wedge$, but using the same nets we also get:

- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \to W_0) = \frac{1}{2} < 1 = C_{\mathrm{seq}}(W_4^{\mathrm{seq}})$,
- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \wedge W_0) = \frac{5}{6} < 1 = C_{\mathrm{seq}}(W_4^{\mathrm{seq}})$,
- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \times W_0) = \frac{5}{7} < 1 = C_{\mathrm{seq}}(W_4^{\mathrm{seq}})$,
- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \circlearrowleft W_0) = \frac{2}{3} < 1 = C_{\mathrm{seq}}(W_4^{\mathrm{seq}})$.
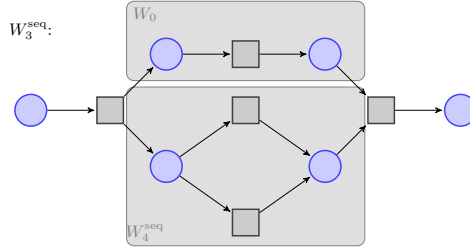

Fig. 37: A workflow net $W_3^{\mathrm{seq}} = W_4^{\mathrm{seq}} \wedge W_0$ with $C_{\mathrm{seq}}(W_3^{\mathrm{seq}}) = 1 - \frac{2}{12} = \frac{5}{6}$, $C_{\mathrm{seq}}(W_0) = 1 - \frac{2}{2} = 0$ and $C_{\mathrm{seq}}(W_4^{\mathrm{seq}}) = 1 - \frac{0}{4} = 1$.

COMP ($\checkmark$): Sequentiality is sensitive to compositions. As an example, Figure 38 shows nets W_4^{seq} and W_5^{seq} with $C_{\mathrm{seq}}(W_4^{\mathrm{seq}}) = 1 = C_{\mathrm{seq}}(W_5^{\mathrm{seq}})$, but for which we get:

- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \to W_0) = \frac{5}{8} \neq \frac{9}{13} = C_{\mathrm{seq}}(W_5^{\mathrm{seq}} \to W_0)$,
- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \wedge W_0) = \frac{5}{6} \neq \frac{15}{17} = C_{\mathrm{seq}}(W_5^{\mathrm{seq}} \wedge W_0)$,
- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \times W_0) = \frac{5}{7} \neq \frac{14}{19} = C_{\mathrm{seq}}(W_5^{\mathrm{seq}} \times W_0)$,
- $C_{\mathrm{seq}}(W_4^{\mathrm{seq}} \circlearrowleft W_0) = \frac{2}{3} \neq \frac{16}{23} = C_{\mathrm{seq}}(W_5^{\mathrm{seq}} \to W_0)$.
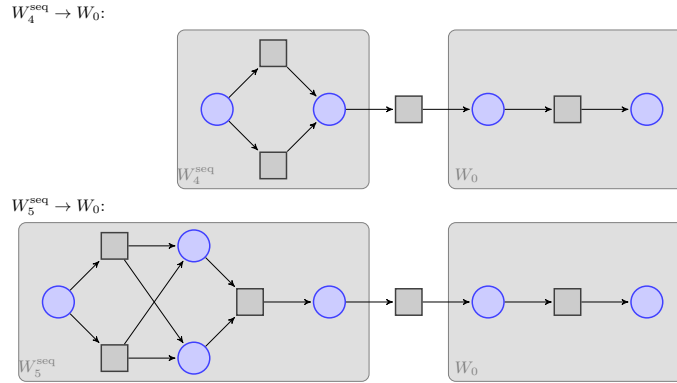

Fig. 38: Example nets W_4^{seq} and W_5^{seq}, each composed sequentially with W_0.

PERM ($\checkmark$): Figure 39 shows two workflow nets, W_6^{seq} and W_7^{seq}, that are permutations of each other, but $C_{\mathrm{seq}}(W_6^{\mathrm{seq}}) = 1 \neq \frac{7}{8} = C_{\mathrm{seq}}(W_7^{\mathrm{seq}})$.
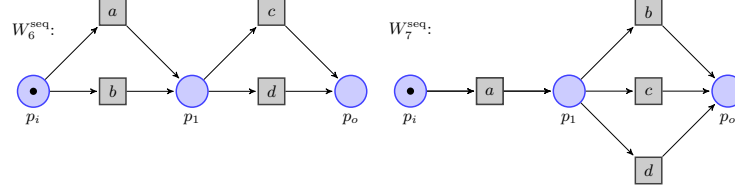

Fig. 39: Two workflow nets, W_6^{seq} and W_7^{seq}, with $W_6^{\mathrm{seq}} \in Perm(W_7^{\mathrm{seq}})$, but $C_{\mathrm{seq}}(W_6^{\mathrm{seq}}) = 1$ and $C_{\mathrm{seq}}(W_7^{\mathrm{seq}}) = \frac{7}{8}$.

ROB$_\ell$ ($\checkmark$): The C_{seq} measure is robust against relabellings, as can be seen straighforwardly by its insensitivity to transition labels generally.

¬SUB ($\checkmark$): The smallest possible workflow net is W_0, and $C_{\mathrm{seq}}(W_0) = 0$ as it has no connectors. We can see that $C_{\mathrm{seq}}(W_0, \oplus W_0) > 0 = C_{\mathrm{seq}}(W_0) + C_{\mathrm{seq}}(W_0)$ for $\oplus \in \{\wedge, \times, \circlearrowright\}$, since these operations introduce new connector nodes and thereby elevate the complexity of the net above 0.

However, C_{seq} is subadditive for the $\rightarrow$ operator. Let $M_1, M_2 \in \mathcal{M}$ be two nets with $M_1 = (P_1, T_1, F_1, \ell_1, p_i^1, p_o^1)$ and $M_2 = (P_2, T_2, F_2, \ell_2, p_i^2, p_o^2)$. $\rightarrow$ introduces two new arcs, which can both have connector-endpoints, if p_i^1 and p_i^2 are both connectors. Therefore, if we take s_1 as the number of sequential arcs in M_1 and s_2 the number of sequential arcs in M_2, we get:

$$C_{\mathrm{seq}}(M_1 \rightarrow M_2)) \leq \frac{s_1 + s_2 + 2}{|F_1| + |F_2| + 2} \leq \frac{s_1 + s_2}{|F_1| + |F_2|} \leq \frac{s_1}{|F_1|} + \frac{s_2}{|F_2|}$$

since $s_1 \leq |F_1|$ and $s_2 \leq |F_2|$. Thus, for the sequence operator, we get $C_{\mathrm{seq}}(M_1 \rightarrow M_2) \leq C_{\mathrm{seq}}(M_1) + C_{\mathrm{seq}}(M_2)$. The associativity of the $\rightarrow$ operation means the claim also holds if we increase the number of workflow nets composed in sequence.

DEF ($\checkmark$): Workflow nets always have at least two arcs, so the denominator of C_{seq} can never be 0. Furthermore, $C_{\mathrm{seq}}(W) \geq 0$ for all workflow nets W, as obviously $|\{(x, y) \in F \mid x, y \notin C^W\}| \leq |F|$.

MIN ($\checkmark$): The minimum score of C_{seq} is 0, which can be reached for the smallest possible workflow net W_0. In this net, there are no connector nodes, so all arcs go from a non-connector node to another non-connector node.

INF ($\checkmark$): Figure 36 shows how to construct infinitely many workflow nets of different complexity.

¬SUP ($\checkmark$): The counterexamples for the property MON also show that C_{seq} is not superadditive with respect to any of the operations.

ADD ($\times$): The counterexamples for the property MON also show that C_{seq} is not additive with respect to any of the operations.

Average Connector Degree The average connector degree is the number of arcs entering or leaving a connector node divided by the total number of connectors [3, pp.120-121]. If this amount is high for a workflow net, it is less understandable than a workflow net with low average connector degree.

$$C_{\mathrm{acd}}(W) = \frac{1}{|\mathcal{C}^W|} \cdot \sum_{x \in \mathcal{C}^W} \deg(x) \tag{14}$$

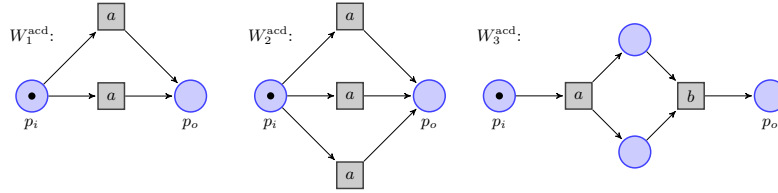Figure 40 shows three example nets and their respective complexity scores.

Fig. 40: Three workflow nets, $W_1^{\mathrm{acd}}, W_2^{\mathrm{acd}}, W_3^{\mathrm{acd}}$, with $C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) = 2$ and $C_{\mathrm{acd}}(W_2^{\mathrm{acd}}) = C_{\mathrm{acd}}(W_3^{\mathrm{acd}}) = 3$.

¬**TRIV** (✓): For the two nets W_1^{acd} and W_2^{acd} of Figure 40, we get the complexity scores $C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) = 2 \neq 3 = C_{\mathrm{acd}}(W_2^{\mathrm{acd}})$.

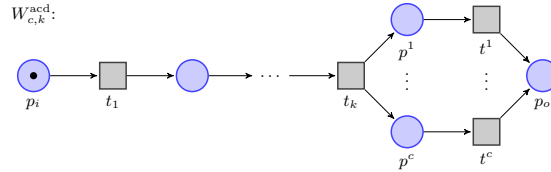**FIN** (×): Follows directly by Theorem 1. Figure 41 shows how to construct such a sequence of places and transitions.

Fig. 41: A workflow net $W_{c,k}^{\mathrm{acd}}$ with $k+c$ transitions and $k+c+1$ places for any $c, k \in \mathbb{N}$ with $c \geq 2$. Its complexity score is $C_{\mathrm{acd}}(W_{c,k}^{\mathrm{acd}}) = \frac{2c+1}{2} = c + \frac{1}{2}$.

COL (✓): The workflow nets W_2^{acd} and W_3^{acd} of Figure 40 are different in structure, but have equal complexity: $C_{\mathrm{acd}}(W_2^{\mathrm{acd}}) = C_{\mathrm{acd}}(W_3^{\mathrm{acd}}) = 3$.

IND$_L$ (✓): The two workflow nets W_1^{acd} and W_2^{acd} of Figure 40 have the same language $L(W_1^{\mathrm{acd}}) = \{\varepsilon, a\} = L(W_2^{\mathrm{acd}})$, but receive different complexity scores $C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) = 2 \neq 3 = C_{\mathrm{acd}}(W_2^{\mathrm{acd}})$.

MON (×): Consider the workflow nets W_5^{acd} and W_6^{acd} of Figure 42. We get:
  − $C_{\mathrm{acd}}(W_6^{\mathrm{acd}} \to W_5^{\mathrm{acd}}) = \frac{14}{4} = 3.5 > 2 = C_{\mathrm{acd}}(W_6^{\mathrm{acd}})$,

- $C_{\mathrm{acd}}(W_6^{\mathrm{acd}} \wedge W_5^{\mathrm{acd}}) = \frac{22}{6} = 3.\overline{6} > 2 = C_{\mathrm{acd}}(W_6^{\mathrm{acd}})$,
- $C_{\mathrm{acd}}(W_6^{\mathrm{acd}} \times W_5^{\mathrm{acd}}) = \frac{20}{6} = 3.\overline{3} > 2 = C_{\mathrm{acd}}(W_6^{\mathrm{acd}})$,
- $C_{\mathrm{acd}}(W_6^{\mathrm{acd}} \circlearrowleft W_5^{\mathrm{acd}}) = \frac{22}{6} = 3.\overline{6} > 2 = C_{\mathrm{acd}}(W_6^{\mathrm{acd}})$.
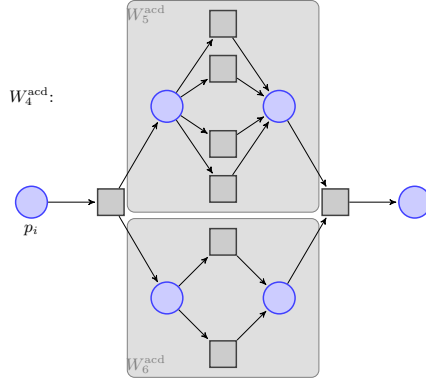
Fig. 42: A workflow net $W_4^{\mathrm{acd}} = W_6^{\mathrm{acd}} \wedge W_5^{\mathrm{acd}}$ with $C_{\mathrm{acd}}(W_4^{\mathrm{acd}}) = \frac{22}{6} = 3.\overline{6}$, $C_{\mathrm{acd}}(W_5^{\mathrm{acd}}) = 4$ and $C_{\mathrm{acd}}(W_6^{\mathrm{acd}}) = 2$.

COMP ($\checkmark$): The workflow nets W_2^{acd} and W_3^{acd} of Figure 40 have the same complexity score $C_{\mathrm{acd}}(W_2^{\mathrm{acd}}) = 3 = C_{\mathrm{acd}}(W_3^{\mathrm{acd}})$. We get:
  - $C_{\mathrm{acd}}(W_2^{\mathrm{acd}} \to W_2^{\mathrm{acd}}) = \frac{14}{4} = 3.5 \neq 3.25 = \frac{13}{4} = C_{\mathrm{acd}}(W_3^{\mathrm{acd}} \to W_2^{\mathrm{acd}})$,
  - $C_{\mathrm{acd}}(W_2^{\mathrm{acd}} \wedge W_2^{\mathrm{acd}}) = \frac{22}{6} = 3.\overline{6} \neq 3.\overline{3} = \frac{20}{6} = C_{\mathrm{acd}}(W_3^{\mathrm{acd}} \wedge W_2^{\mathrm{acd}})$,
  - $C_{\mathrm{acd}}(W_2^{\mathrm{acd}} \times W_2^{\mathrm{acd}}) = \frac{20}{6} = 3.\overline{3} \neq 3 = \frac{18}{6} = C_{\mathrm{acd}}(W_3^{\mathrm{acd}} \times W_2^{\mathrm{acd}})$,
  - $C_{\mathrm{acd}}(W_2^{\mathrm{acd}} \circlearrowleft W_2^{\mathrm{acd}}) = \frac{22}{6} = 3.\overline{6} \neq 3 = \frac{18}{6} = C_{\mathrm{acd}}(W_3^{\mathrm{acd}} \circlearrowleft W_2^{\mathrm{acd}})$.

PERM ($\checkmark$): Figure 43 shows two workflow nets, W_7^{acd} and W_8^{acd}, that are permutations of each other, but $C_{\mathrm{acd}}(W_7^{\mathrm{acd}}) = 2.\overline{6} \neq 3.5 = C_{\mathrm{acd}}(W_8^{\mathrm{acd}})$.
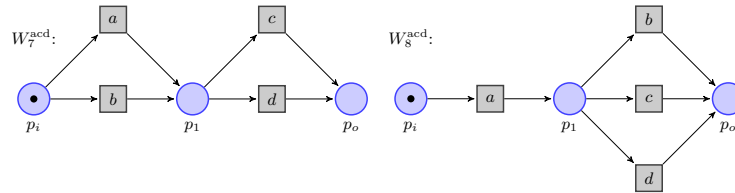

Fig. 43: Two workflow nets, W_7^{acd} and W_8^{acd}, with $W_8^{\mathrm{acd}} \in Perm(W_7^{\mathrm{acd}})$, $C_{\mathrm{acd}}(W_7^{\mathrm{acd}}) = 2.\overline{6}$ and $C_{\mathrm{acd}}(W_8^{\mathrm{acd}}) = 3.5$.

ROB$_\ell$ ($\checkmark$): The C_{acd} measure is robust against relabellings, as can be seen by its general insensitivity to transition labels.

¬SUB ($\times$): If we consider only inputs workflow nets where C_{acd} is defined, and thus can assume that all input workflow nets have at least one connector, C_{acd} is subadditive for the operations $\to$, $\wedge$, $\times$ and $\circlearrowleft$.

Proof. Let $M_1 = (P^1, T^1, F^1, p_i^1, p_o^1)$ and $M_2 = (P^2, T^2, F^2, p_i^2, p_o^2)$ be two workflow nets. The sequential composition $M_1 \to M_2$ introduces only one new transition with one incoming and one outgoing arc. By definition, this transition can't be a connector node, since neither its number of incoming nor the number of outgoing arcs exceeds 1. The degrees of p_o^1 and p_i^2 increase by 1 with this operation. This might impact the measure when p_o^1 or p_i^2 are connector nodes. With $C_{\mathrm{acd}}(M_1) = \frac{d_1}{|\mathcal{C}_1^M|}$ and $C_{\mathrm{acd}}(M_2) = \frac{d_2}{|\mathcal{C}_2^M|}$, we have:

$$C_{\mathrm{acd}}(M_1 \to M_2) \leq \frac{d_1 + d_2 + 2}{|\mathcal{C}_1^M| + |\mathcal{C}_2^M|}.$$

Since $\frac{d_1 + d_2 + 2}{|\mathcal{C}_1^M| + |\mathcal{C}_2^M|} \leq \frac{|\mathcal{C}_2^M| \cdot d_1 + |\mathcal{C}_1^M| \cdot d_2}{|\mathcal{C}_1^M||\mathcal{C}_2^M|}$, for $d_1, d_2 \geq 2$ and $|\mathcal{C}_1^M|, |\mathcal{C}_2^M| \geq 2$:

$$C_{\mathrm{acd}}(M_1 \to M_2) \leq \frac{d_1}{|\mathcal{C}_1^M|} + \frac{d_2}{|\mathcal{C}_2^M|}$$
$$C_{\mathrm{acd}}(M_1 \to M_2) \leq C_{\mathrm{acd}}(M_1) + C_{\mathrm{acd}}(M_2).$$

As sequential composition is associative, this claim also holds if we increase the amount of workflow nets composed by the $\to$-operator.

The operators $\wedge$, $\times$ and $\circlearrowleft$ each create two new connectors of degree at least n for their n parameters $M_1, \ldots, M_n$. They may also increase the node degree of the initial and final places of these nets by one, if they are connectors. With $C_{\mathrm{acd}}(M_i) = \frac{d_i}{|\mathcal{C}_i^{M_i}|}$ for $i \in \{1, \ldots, n\}$, we get:

$$C_{\mathrm{acd}}(\oplus(M_1, \ldots, M_n)) \leq \frac{4(n+1) + \sum_{i=1}^n d_i}{2 + \sum_{i=1}^n |\mathcal{C}_i^{M_i}|}.$$

When there are two nets, each with the minimum possible average connector degree 2, this bound is $\frac{20}{6} = 3.\overline{3}$, which is less than the sum of the average degrees, 4. The sum of $C_{\mathrm{acd}}(M_i)$ diverges as n, d_i and $|C_i^{M_i}|$ increase. Therefore, we have

$$C_{\mathrm{acd}}(\oplus(M_1, \ldots, M_n)) \leq C_{\mathrm{acd}}(M_1) + \cdots + C_{\mathrm{acd}}(M_2)$$

$\square$

DEF ($\times$): When there are no connectors in a workflow net, we would divide by 0, leading to an undefined score. Figure 2 shows such a workflow net without connectors. If we define C_{acd} to be 0 in this case, the measure only returns non-negative values, since $\deg(x) \geq 0$ for each node x in a workflow net.

MIN ($\checkmark$): The smallest possible value of C_{acd} (without introducing the special case above) is 2, since a connector needs at least 2 outgoing or 2 incoming arcs to classify as a connector. The workflow net W_1^{acd} of Figure 40 is the workflow net with the smallest average connector degree and receives this complexity score.

INF ($\checkmark$): The net $W_{c,k}^{\mathrm{acd}}$ in Figure 41 shows how to construct nets with an average connector degree $\frac{2c+1}{2}$ where $c \in \mathbb{N}$ with $c \geq 2$. Therefore, we get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{acd}}(M) = c\}| \geq \{c \in \mathbb{N} \mid c \geq 2\}| = \infty$.

¬SUP ($\checkmark$): Take the workflow nets W_1^{acd} and W_2^{acd} of Figure 40. We get:
  - $C_{\mathrm{acd}}(W_1^{\mathrm{acd}} \to W_2^{\mathrm{acd}}) = 3 < 5 = 2 + 3 = C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) + C_{\mathrm{acd}}(W_2^{\mathrm{acd}})$,
  - $C_{\mathrm{acd}}(W_1^{\mathrm{acd}} \wedge W_2^{\mathrm{acd}}) = 3.\overline{3} < 5 = 2 + 3 = C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) + C_{\mathrm{acd}}(W_2^{\mathrm{acd}})$,
  - $C_{\mathrm{acd}}(W_1^{\mathrm{acd}} \times W_2^{\mathrm{acd}}) = 3 < 5 = 2 + 3 = C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) + C_{\mathrm{acd}}(W_2^{\mathrm{acd}})$,
  - $C_{\mathrm{acd}}(W_1^{\mathrm{acd}} \circlearrowright W_2^{\mathrm{acd}}) = 3.\overline{3} < 5 = 2 + 3 = C_{\mathrm{acd}}(W_1^{\mathrm{acd}}) + C_{\mathrm{acd}}(W_2^{\mathrm{acd}})$.

ADD ($\times$): The counter-examples for property ¬SUP also show that C_{acd} is not additive.

5.3 Path Complexity

Complexity measures of the PATH COMPLEXITY dimension take the complexity of paths in the model into account. The model is interpreted as a directed graph and paths are defined as in graph-theory, ignoring all semantics of choice or concurrent connectors. Thus, these measures ignore the semantics of the model and rate the complexity based on how understandable the underlying graph is.

Depth For a workflow net $W = (P, T, F, \ell, p_i, p_o)$, the depth [3, pp.124-125] of a node $v \in P \cup T$ is defined by its in-depth and its out-depth. The in-depth counts how many split nodes precede v that were not closed by a join before v, while the out-depth counts how many join nodes succeed v that aren't opened by a split after v. For the in-depth, we inspect all paths from p_i to v. Let $\mathcal{P}_{p_i,v}$ be the set of all such paths that don't revisit any nodes. Further, let $\mathcal{S}^W := \mathcal{S}_{\mathrm{and}}^W \cup \mathcal{S}_{\mathrm{xor}}^W$ the set of all split nodes in W and $\mathcal{J}^W := \mathcal{J}_{\mathrm{and}}^W \cup \mathcal{J}_{\mathrm{xor}}^W$ the set of all join nodes in W. For $p = (v_1, \ldots, v_n) \in \mathcal{P}_{p_i,v}$, we define inductively:

$$\lambda_W(v_1) = \lambda_W(p_i) := 0 \tag{15}$$

$$\lambda_p(v_n) := \begin{cases} \lambda_W(v_{n-1}) + 1 & \text{if } v_{n-1} \in \mathcal{S}^W \wedge v_n \notin \mathcal{J}^W \\ \lambda_W(v_{n-1}) & \text{if } v_{n-1} \in \mathcal{S}^W \wedge v_n \in \mathcal{J}^W \\ \lambda_W(v_{n-1}) & \text{if } v_{n-1} \notin \mathcal{S}^W \wedge v_n \notin \mathcal{J}^W \\ \lambda_W(v_{n-1}) - 1 & \text{if } v_{n-1} \notin \mathcal{S}^W \wedge v_n \in \mathcal{J}^W \end{cases} \tag{16}$$

$$\lambda_W(v) := \max\left\{0, \max_{p \in \mathcal{P}_{p_i,v}} \lambda_p(v)\right\} \quad \text{(for any } v \neq p_i) \tag{17}$$

For the out-depth, we set $\overleftarrow{W} := (P, T, \overleftarrow{F}, \ell, p_o, p_i)$ with $\overleftarrow{F} := \{(u,v) \mid (v,u) \in F\}$ as the workflow net with all arcs reversed. Now, we can reuse the definition of the in-depth to define the out-depth. We define the depth of a workflow net W as the maximum depth of any node in W.

$$C_{\mathrm{depth}}(W) := \max\{\min\{\lambda_W(v), \lambda_{\overleftarrow{W}}(v)\} \mid v \in P \cup T\} \tag{18}$$

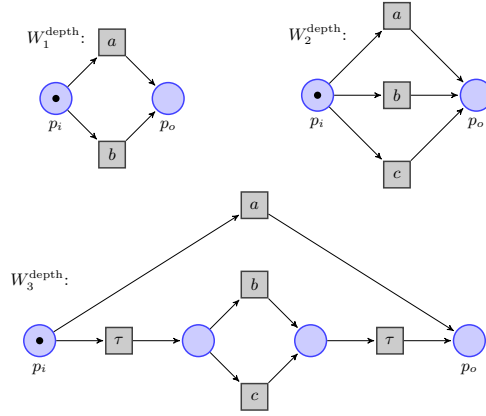Figure 44 shows three example nets and their respective complexity scores.

Fig. 44: Three workflow nets, W_1^{depth}, W_2^{depth}, W_3^{depth}, with $C_{\text{depth}}(W_3^{\text{depth}}) = 2$ and $C_{\text{depth}}(W_1^{\text{depth}}) = C_{\text{depth}}(W_2^{\text{depth}}) = 1$.

¬**TRIV** (✓): For the two nets W_2^{depth} and W_3^{depth} of Figure 44, we get the complexity scores $C_{\text{depth}}(W_2^{\text{depth}}) = 1 \neq 2 = C_{\text{depth}}(W_3^{\text{depth}})$.

FIN (×): There are infinitely many workflow nets with complexity 1: Take the workflow net W_2^{depth} of Figure 44. If we add more transitions t with $^\bullet t = \{p_i\}$ and $t^\bullet = \{p_o\}$ to this net, we don't affect the depth of the net. Therefore, all such workflow nets get the same complexity score, 1.

COL (✓): The workflow nets W_1^{depth} and W_2^{depth} are different in structure, but both get the complexity score $C_{\text{depth}}(W_1^{\text{depth}}) = 1 = C_{\text{depth}}(W_2^{\text{depth}})$.

IND$_L$ (✓): Take the workflow nets W_2^{depth} and W_3^{depth} of Figure 44. Their languages are $L(W_2^{depth}) = \{\varepsilon, a, b, c\} = L(W_3^{depth})$, but their complexity scores are $C_{\text{depth}}(W_2^{\text{depth}}) = 1 \neq 2 = C_{\text{depth}}(W_3^{\text{depth}})$.

MON (✓): Let $M_1, M_2 \in \mathcal{M}$ be workflow nets. By definition, the depth of composed nets can be computed as follows:

– $C_{\text{depth}}(M_1 \wedge M_2) = 1 + \max\{C_{\text{depth}}(M_1), C_{\text{depth}}(M_2)\}$,

– $C_{\text{depth}}(M_1 \times M_2) = 1 + \max\{C_{\text{depth}}(M_1), C_{\text{depth}}(M_2)\}$,

– $C_{\text{depth}}(M_1 \circlearrowleft M_2) = \max\{C_{\text{depth}}(M_1), C_{\text{depth}}(M_2) + 1\}$.

Therefore, $C_{\text{depth}}(M_1 \oplus M_2) \geq C_{\text{depth}}(M_1)$ for any $\oplus \in \{\wedge, \times, \circlearrowleft\}$. Consider the net $M = M_1 \rightarrow M_2$. Further let v, w be two nodes of M such that $\min\{\lambda_{M_1}(v), \lambda_{\overleftarrow{M_1}}(v)\} = C_{\text{depth}}(M_1)$, i.e. v is the deepest node of the net M_1, and $\min\{\lambda_{M_2}(w), \lambda_{\overleftarrow{M_2}}(w)\} = C_{\text{depth}}(M_2)$. Then, we have $\lambda_M(v) = \lambda_{M_1}(v)$,

$\lambda_M(w) \geq \lambda_{W_2}(w)$, $\lambda_{\overleftarrow{M}}(w) = \lambda_{\overleftarrow{M_2}}(w)$ and $\lambda_{\overleftarrow{M}}(v) \geq \lambda_{\overleftarrow{M_1}}(v)$. We deduce:

$$
\begin{aligned}
C_{\mathrm{depth}}(M) &= \max_v\{\min\{\lambda_M(v), \lambda_{\overleftarrow{M}}(v)\}\} \\
&= \min\{\max\{\lambda_M(v), \lambda_M(w)\}, \max\{\lambda_{\overleftarrow{M}}(w), \lambda_{\overleftarrow{M}}(v)\}\} \\
&\geq \max\{\min\{\lambda_{M_1}(v), \lambda_{\overleftarrow{M_1}}(v)\}, \min\{\lambda_{M_2}(w), \lambda_{\overleftarrow{M_2}}(w)\}\}\} \\
&= \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_2)\} \\
&\geq C_{\mathrm{depth}}(M_1)
\end{aligned}
$$

Therefore, C_{depth} is monotone for all operations of Definition 6.

COMP ($\times$): The property does not hold for operations $\oplus \in \{\wedge, \times, \circlearrowleft\}$. Let $M_1, M_2, M_3 \in \mathcal{M}$ be workflow nets with $C_{\mathrm{depth}}(M_1) = C_{\mathrm{depth}}(M_2)$. We get:

$$
\begin{aligned}
- \quad C_{\mathrm{depth}}(M_1 \wedge M_3) &= 1 + \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_3)\} \\
&= 1 + \max\{C_{\mathrm{depth}}(M_2), C_{\mathrm{depth}}(M_3)\} \\
&= C_{\mathrm{depth}}(M_2 \wedge M_3), \\
- \quad C_{\mathrm{depth}}(M_1 \times M_3) &= 1 + \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_3)\} \\
&= 1 + \max\{C_{\mathrm{depth}}(M_2), C_{\mathrm{depth}}(M_3)\} \\
&= C_{\mathrm{depth}}(M_2 \times M_3), \\
- \quad C_{\mathrm{depth}}(M_1 \circlearrowleft M_3) &= \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_3) + 1\} \\
&= \max\{C_{\mathrm{depth}}(M_2), C_{\mathrm{depth}}(M_3) + 1\} \\
&= C_{\mathrm{depth}}(M_2 \wedge M_3).
\end{aligned}
$$

For the operator $\rightarrow$, consider the net W_4^{depth} of Figure 45 and the net $\overleftarrow{W_4^{\mathrm{depth}}}$ obtained by reversing the arcs of W_4^{depth}. Both workflow nets have complexity $C_{\mathrm{depth}}(W_4^{\mathrm{depth}}) = 1 = C_{\mathrm{depth}}(\overleftarrow{W_4^{\mathrm{depth}}})$, but composing these nets with W_4^{depth} gives $C_{\mathrm{depth}}(\overleftarrow{W_4^{\mathrm{depth}}} \rightarrow W_4^{\mathrm{depth}}) = 2 \neq 1 = C_{\mathrm{depth}}(W_4^{\mathrm{depth}} \oplus W_4^{\mathrm{depth}})$.
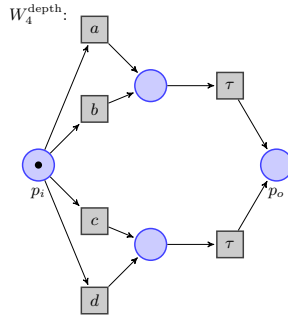
Fig. 45: A workflow net W_4^{depth} with $C_{\mathrm{depth}}(W_4^{\mathrm{depth}}) = 1 = C_{\mathrm{depth}}(\overleftarrow{W_4^{\mathrm{depth}}})$.

PERM ($\checkmark$): Figure 46 shows two workflow nets, W_5^{depth} and W_6^{depth}, that are permutations of each other, but $C_{\mathrm{depth}}(W_5^{\mathrm{depth}}) = 2 \neq 1 = C_{\mathrm{depth}}(W_6^{\mathrm{depth}})$.
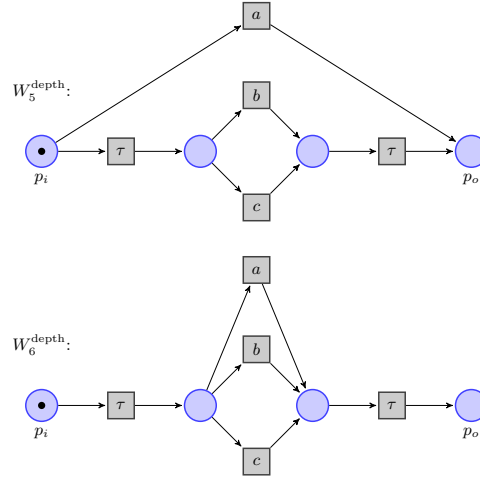
Fig. 46: Two workflow nets with $W_5^{\mathrm{depth}} \in Perm(W_6^{\mathrm{depth}})$ and complexity scores $C_{\mathrm{depth}}(W_5^{\mathrm{depth}}) = 2 \neq 1 = C_{\mathrm{depth}}(W_6^{\mathrm{depth}})$.

ROB$_\ell$ (✓): A relabelling of the transitions does not affect C_{depth}.

¬SUB (✓): Consider the workflow net W_0 of Figure 2. For all operators except $\rightarrow$, we get:

- $C_{\mathrm{depth}}(W_0 \wedge W_0) = 1 > 0 = 0 + 0 = C_{\mathrm{depth}}(W_0) + C_{\mathrm{depth}}(W_0)$,
- $C_{\mathrm{depth}}(W_0 \times W_0) = 1 > 0 = 0 + 0 = C_{\mathrm{depth}}(W_0) + C_{\mathrm{depth}}(W_0)$,
- $C_{\mathrm{depth}}(W_0 \circlearrowleft W_0) = 1 > 0 = 0 + 0 = C_{\mathrm{depth}}(W_0) + C_{\mathrm{depth}}(W_0)$.

Next, we consider the operator $\rightarrow$. For each $n \in \mathbb{N}$ there is a workflow net $N := W_{sub,n}^{\mathrm{depth}}$ with $\max_v\{\lambda_N(v)\} = 1$, $\max_v\{\lambda_{\overleftarrow{N}}(v)\} = n$ and, for all $v \in P \cup T$, $\lambda_N(v) = 1 \Leftrightarrow \lambda_{\overleftarrow{N}}(v) = n$. Figure 47 shows a possible construction for $n = 3$. For any $n > 2$, we get that $C_{\mathrm{depth}}(N) = C_{\mathrm{depth}}(\overleftarrow{N}) = 1$ and $C_{\mathrm{depth}}(\overleftarrow{N} \rightarrow N) = n > 2 = C_{\mathrm{depth}}(N) + C_{\mathrm{depth}}(\overleftarrow{N})$.

DEF (✓): C_{depth} is well-defined and deterministic, since $\mathcal{P}_{p_i,v}$ is finite for each node $v \in P \cup T$ and because each node lies on a path from p_i to p_o in a workflow net. By definition of $\lambda_W(v)$, C_{depth} returns only non-negative values.

MIN (✓): The smallest possible workflow net W_0 of Figure 2 has no splits and joins. Thus, $C_{\mathrm{depth}}(W_0) = 0$, which is the minimum of this complexity measure.

INF (✓): Let $n \in \mathbb{N}$ with $n \geq 3$. We can easily construct a workflow net with complexity n by creating n subsequent split connectors followed by n subsequent join connectors on a path from p_i to p_o. Therefore, we get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{depth}}(M) = c\}| \geq |\{n \in \mathbb{N} \mid n \geq 3\}| = \infty$.

¬SUP (✓): For the workflow nets W_1^{depth} of Figure 44, we can easily see that $C_{\mathrm{depth}}(W_1^{\mathrm{depth}} \rightarrow W_1^{\mathrm{depth}}) = 1 < 2 = C_{\mathrm{depth}}(W_1^{\mathrm{depth}}) + C_{\mathrm{depth}}(W_1^{\mathrm{depth}})$. For all other operations, we use our analyses for the property MON and deduce:
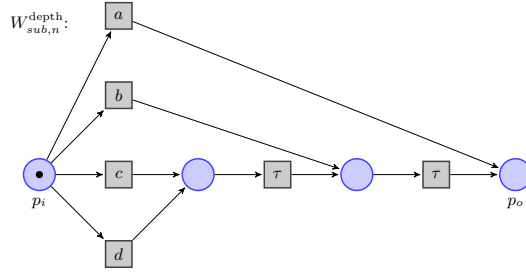
Fig. 47: A workflow net $W_{sub,n}^{\mathrm{depth}}$ with $C_{\mathrm{depth}}(W_{sub,n}^{\mathrm{depth}}) = 1 = C_{\mathrm{depth}}(\overleftarrow{W_{sub,n}^{\mathrm{depth}}})$, but where $C_{\mathrm{depth}}(\overleftarrow{W_{sub,n}^{\mathrm{depth}}} \to W_{sub,n}^{\mathrm{depth}})) = 3$.

- $C_{\mathrm{depth}}(M_1 \wedge M_2) = 1 + \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_2)\}$
  $< C_{\mathrm{depth}}(M_1) + C_{\mathrm{depth}}(M_2),$
- $C_{\mathrm{depth}}(M_1 \times M_2) = 1 + \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_2)\}$
  $< C_{\mathrm{depth}}(M_1) + C_{\mathrm{depth}}(M_2),$
- $C_{\mathrm{depth}}(M_1 \circlearrowleft M_2) = \max\{C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_2) + 1\}$
  $< C_{\mathrm{depth}}(M_1) + C_{\mathrm{depth}}(M_2)$

for workflow nets $M_1, M_2 \in \mathcal{M}$ with $C_{\mathrm{depth}}(M_1), C_{\mathrm{depth}}(M_2) \geq 2$.

ADD (×): The counter-examples for property ¬SUB also show that C_{depth} is not additive.

Diameter The diameter [3, p.119] of a workflow net $W = (P, T, F, \ell, p_i, p_o)$ is the maximal length of a way from p_i to p_o where no arc is used more than once. Note that, in contrast to paths, ways are allowed to revisit nodes. The length of a way $w = v_1, \ldots, v_k$ it the number of visited nodes, where revisited nodes can be counted more than once. For $w = v_1, \ldots, v_k$, we have $|w| = k$. Let $\mathcal{W}_{p_i, p_o}$ be the set of all ways as defined above. We define:

$$C_{\mathrm{diam}}(W) = \max\{|p| \mid p \in \mathcal{W}_{p_i, p_o}\}. \tag{19}$$

Figure 48 shows three example nets and their respective complexity scores.

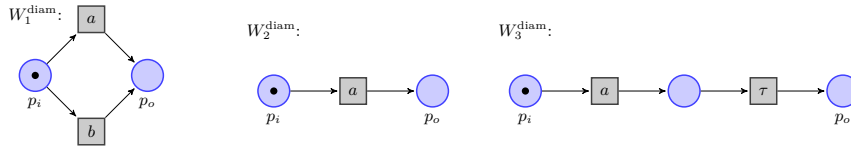Fig. 48: Three workflow nets, $W_1^{\mathrm{diam}}, W_2^{\mathrm{diam}}, W_3^{\mathrm{diam}}$, with $C_{\mathrm{diam}}(W_3^{\mathrm{diam}}) = 5$ and $C_{\mathrm{diam}}(W_1^{\mathrm{diam}}) = C_{\mathrm{diam}}(W_2^{\mathrm{diam}}) = 3$.

¬TRIV (✓): For the two nets W_2^{diam} and W_3^{diam} of Figure 48, we get the complexity scores $C_{\text{diam}}(W_2^{\text{diam}}) = 3 \neq 5 = C_{\text{diam}}(W_3^{\text{diam}})$.

FIN (×): There are infinitely many workflow nets with complexity 3: Take the workflow net W_1^{diam} of Figure 48. If we add more transitions t with ${}^\bullet t = \{p_i\}$ and $t^\bullet = \{p_o\}$ to this net, we don't affect the diameter of the net. Therefore, all such workflow nets get the same complexity score 3.

COL (✓): The workflow nets W_1^{diam} and W_2^{diam} of Figure 48 are different in structure, but both have the same diameter and therefore get the same complexity score $C_{\text{diam}}(W_1^{\text{diam}}) = 3 = C_{\text{diam}}(W_2^{\text{diam}})$.

IND$_L$ (✓): Take the workflow nets W_2^{diam} and W_3^{diam} of Figure 48. Their languages are $L(W_2^{\text{diam}}) = \{\varepsilon, a\} = L(W_3^{\text{diam}})$, but their complexity scores are $C_{\text{diam}}(W_2^{\text{diam}}) = 3 \neq 5 = C_{\text{diam}}(W_3^{\text{diam}})$.

MON (✓): We use the following Theorem:

Theorem 7. *Let $M_1, M_2 \in \mathcal{M}$ be workflow nets. Then,*
- *$C_{diam}(M_1 \to M_2) = 1 + C_{diam}(M_1) + C_{diam}(M_2)$,*
- *$C_{diam}(M_1 \wedge M_2) = 4 + \max\{C_{diam}(M_1), C_{diam}(M_2)\}$,*
- *$C_{diam}(M_1 \times M_2) = 4 + \max\{C_{diam}(M_1), C_{diam}(M_2)\}$,*
- *$C_{diam}(M_1 \circlearrowright M_2) = 8 + C_{diam}(M_1)$.*

Proof. We show these claims for each operation individually. Let w_1 be the longest way in M_1 and w_2 be the longest way in M_2.
- Let t^* be the transition introduced by the $\to$ operation. The longest way in $M_1 \to M_2$ then is w_1, t^*, w_2. Its length is $|w_1| + |w_2| + 1$.
- Let $p_i^*, t_i^*, t_o^*, p_o^*$ be the places and transitions introduced by the $\wedge$ operation. The longest way in $M_1 \wedge M_2$ then is $p_i^*, t_i^*, w, t_o^*, p_o^*$, where w is the longest of the paths w_1, w_2. Its length is $\max\{|w_1|, |w_2|\} + 4$.
- Let $p_i^*, t_1^*, t_2^*, s_1^*, s_2^*, p_o^*$ be the places and transitions introduced by the $\times$ operation. The longest way in $M_1 \times M_2$ then is $p_i^*, t_m^*, w_m^*, s_m^*, p_o^*$, where m is the index of the longest of the paths w_1, w_2. Its length is $\max\{|w_1|, |w_2|\} + 4$.
- Let $p_i^*, t^*, p^*, t_1^*, t_2^*, s_1^*, s_2^*, q^*, s^*, p_o^*$ be the places and transitions introduced by the $\circlearrowright$ operation. The longest way in $M_1 \circlearrowright M_2$ is the way $p_i^*, t^*, p^*, t_1^*, w_1, s_1^*, q^*, s^*, p_o^*$, since we can't visit any nodes of M_2 without needing to reuse arcs of M_1. The length of this path is $|w_1| + 8$. □

With Theorem 7, we get for any $M_1, M_2 \in \mathcal{M}$:
- $C_{\text{diam}}(M_1 \to M_2) = 1 + C_{\text{diam}}(M_1) + C_{\text{diam}}(M_2) > C_{\text{diam}}(M_1)$,
- $C_{\text{diam}}(M_1 \wedge M_2) = 4 + \max\{C_{\text{diam}}(M_1), C_{\text{diam}}(M_2)\} > C_{\text{diam}}(M_1)$,
- $C_{\text{diam}}(M_1 \times M_2) = 4 + \max\{C(M_1, C_{\text{diam}}(M_2)\} > C_{\text{diam}}(M_1)$,
- $C_{\text{diam}}(M_1 \circlearrowright M_2) = 8 + C_{\text{diam}}(M_1) > C_{\text{diam}}(M_1)$.

Note that monotonicity only applies for $\circlearrowright$ because of the way we defined the proposition and the operation: While $C_{\text{diam}}(M_1 \circlearrowright M_2) > C_{\text{diam}}(M_1)$ is always true, $C_{\text{diam}}(M_1 \circlearrowright M_2) > C_{\text{diam}}(M_2)$ is not.

COMP (×): Let $M_1, M_2, M_3 \in \mathcal{M}$ be workflow nets with complexity scores $C_{\text{diam}}(M_1) = C_{\text{diam}}(M_2)$. Theorem 7 gives:

- $C_{\text{diam}}(M_1 \to M_3) = 1 + C_{\text{diam}}(M_1) + C_{\text{diam}}(M_3)$
$$= 1 + C_{\text{diam}}(M_2) + C_{\text{diam}}(M_3) = C_{\text{diam}}(M_2 \to M_3),$$
- $C_{\text{diam}}(M_1 \wedge M_3) = 4 + \max\{C_{\text{diam}}(M_1), C_{\text{diam}}(M_3)\}$
$$= 4 + \max\{C_{\text{diam}}(M_2), C_{\text{diam}}(M_3)\} = C_{\text{diam}}(M_2 \wedge M_3),$$
- $C_{\text{diam}}(M_1 \times M_3) = 4 + \max\{C(M_1, C_{\text{diam}}(M_3)\}$
$$= 4 + \max\{C_{\text{diam}}(M_2), C_{\text{diam}}(M_3)\} = C_{\text{diam}}(M_2 \times M_3),$$
- $C_{\text{diam}}(M_1 \circlearrowleft M_3) = 8 + C_{\text{diam}}(M_1)$
$$= 8 + C_{\text{diam}}(M_2) = C_{\text{diam}}(M_2 \circlearrowleft M_3).$$

PERM ($\checkmark$): Figure 49 shows two workflow nets, W_4^{diam} and W_5^{diam}, that are permutations of each other, but $C_{\text{diam}}(W_4^{\text{diam}}) = 7 \neq 9 = C_{\text{diam}}(W_5^{\text{diam}})$.
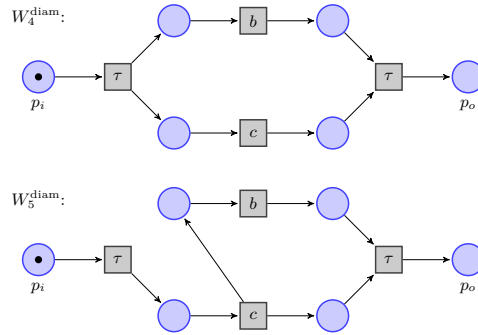

Fig. 49: Two workflow nets with $W_4^{\text{diam}} \in Perm(W_5^{\text{diam}})$ and complexity scores $C_{\text{diam}}(W_4^{\text{diam}}) = 7 \neq 9 = C_{\text{diam}}(W_5^{\text{diam}})$.

ROB$_\ell$ ($\checkmark$): A relabelling of the transitions does not affect C_{diam}.

¬SUB ($\checkmark$): Take the workflow net W_0 of Figure 2. We get:
- $C_{\text{diam}}(M_1 \to M_2) > C_{\text{diam}}(M_1) + C_{\text{diam}}(M_2)$
for all $M_1, M_2 \in \mathcal{M}$ due to Theorem 7,
- $C_{\text{diam}}(W_0 \wedge W_0) = 7 > 6 = 3 + 3 = C_{\text{diam}}(W_0) + C_{\text{diam}}(W_0)$,
- $C_{\text{diam}}(W_0 \times W_0) = 7 > 6 = 3 + 3 = C_{\text{diam}}(W_0) + C_{\text{diam}}(W_0)$,
- $C_{\text{diam}}(W_0 \circlearrowleft W_0) = 11 > 6 = 3 + 3 = C_{\text{diam}}(W_0) + C_{\text{diam}}(W_0)$

DEF ($\checkmark$): Since the output place must be reachable from the input place, there is at least one path in each workflow net, so the maximum always exists. This maximum is non-negative, since the length of a path can't be negative.

MIN ($\checkmark$): The minimum possible diameter is that of the smallest possible workflow net W_0. This workflow net has diameter 3, using two arcs. It is not possible to use less than two arcs for a path from p_i to p_o in a workflow net, as this would mean that $(p_i, p_o) \in F$, contradicting $F \subseteq (P \times T) \cup (T \times P)$.

INF ($\checkmark$): Let $n \in \mathbb{N}$ with $n \geq 3$ and n being an odd number. Figure 50 shows how to construct a workflow net of complexity n, so we immediately get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\text{diam}}(M) = c\}| \geq |\{n \in \mathbb{N} \mid n \geq 3 \wedge \frac{n}{2} \notin \mathbb{N}\}| = \infty$.
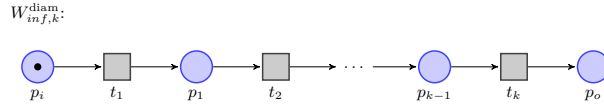
$W_{inf,k}^{\mathrm{diam}}$:

Fig. 50: A workflow net $W_{inf,k}^{\mathrm{diam}}$ with k transitions and $k+1$ places. Its complexity is $C_{\mathrm{diam}}(W_{inf,k}^{\mathrm{diam}}) = 2k + 1$.

¬**SUP** (×): Theorem 7 directly implies that C_{diam} is superadditive with respect to the operation $\rightarrow$, since for any workflow nets $M_1, M_2 \in \mathcal{M}$, we get $C_{\mathrm{diam}}(M_1 \rightarrow M_2) = 1 + C_{\mathrm{diam}}(M_1) + C_{\mathrm{diam}}(M_2) > C_{\mathrm{diam}}(M_1) + C_{\mathrm{diam}}(M_2)$. For the other operations, we find counter-examples for superadditivity: Consider the net $W_{inf,4}^{\mathrm{diam}}$ of Figure 50. We get:

- $C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}} \wedge W_{inf,4}^{\mathrm{diam}}) = 13 < 18 = C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}}) + C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}})$,
- $C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}} \times W_{inf,4}^{\mathrm{diam}}) = 13 < 18 = C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}}) + C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}})$,
- $C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}} \circlearrowleft W_{inf,4}^{\mathrm{diam}}) = 17 < 18 = C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}}) + C_{\mathrm{diam}}(W_{inf,4}^{\mathrm{diam}})$.

ADD (×): The analysis of the property ¬SUB also show that C_{diam} is not additive with respect to any of the operations.

Cyclicity Cycles in a workflow net are often perceived as complex structures. Therefore, a simple process model aims for as many nodes outside of a cycle as possible [3, pp.127-128]. For a workflow net W, cyclicity is the number of nodes that lie on a cycle in W divided by the total number of nodes in W.

$$C_{\mathrm{cyc}}(W) := \frac{|\{x \in P \cup T \mid x \text{ lies on a cycle in } W\}|}{|P| + |T|} \tag{20}$$

Figure 51 shows three example nets and their respective complexity scores.
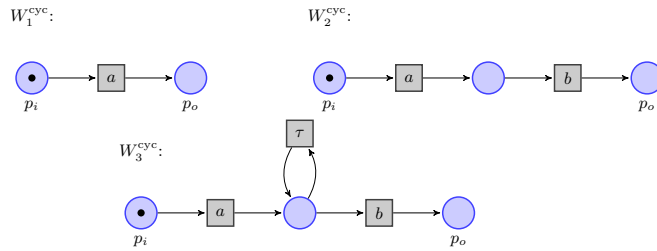
W_1^{cyc}: W_2^{cyc}:

W_3^{cyc}:

Fig. 51: Three workflow nets, W_1^{cyc}, W_2^{cyc}, W_3^{cyc}, with $C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}}) = \frac{1}{3}$ and $C_{\mathrm{cyc}}(W_1^{\mathrm{cyc}}) = C_{\mathrm{cyc}}(W_2^{\mathrm{cyc}}) = 0$.

¬**TRIV** ($\checkmark$): For the two nets W_2^{cyc} and W_3^{cyc} of Figure 51, we get the complexity scores $C_{\mathrm{cyc}}(W_2^{\mathrm{cyc}}) = 0 \neq \frac{1}{3} = C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}})$.

FIN ($\times$): Take the workflow net W_1^{cyc} of Figure 51. If we add more transitions t with $^\bullet t = \{p_i\}$ and $t^\bullet = \{p_o\}$ to this net, we don't change its complexity. Therefore, we can create infinitely many workflow nets with complexity 0.

COL ($\checkmark$): The workflow nets W_1^{cyc} and W_2^{cyc} of Figure 51 are different in structure, but both get the same complexity score $C_{\mathrm{cyc}}(W_1^{\mathrm{cyc}}) = 3 = C_{\mathrm{cyc}}(W_2^{\mathrm{cyc}})$.

IND$_L$ ($\checkmark$): Take the workflow nets W_2^{cyc} and W_3^{cyc} of Figure 29. Their languages are $L(W_2^{cyc}) = \{\varepsilon, a, ab\} = L(W_3^{cyc})$, but their complexity scores are $C_{\mathrm{cyc}}(W_2^{\mathrm{cyc}}) = 0 \neq \frac{1}{3} = C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}})$.

MON ($\times$): Consider the workflow nets W_1^{cyc} and W_3^{cyc} of Figure 51. For the operations $\rightarrow, \times$ and $\circlearrowleft$, we get:

- $C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}} \rightarrow W_1^{\mathrm{cyc}}) = \frac{2}{10} = 0.2 < 0.\overline{3} = \frac{1}{3} = C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}})$,
- $C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}} \wedge W_1^{\mathrm{cyc}}) = \frac{2}{13} \approx 0.1538 < 0.\overline{3} = \frac{1}{3} = C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}})$,
- $C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}} \times W_1^{\mathrm{cyc}}) = \frac{2}{15} = 0.1\overline{3} < 0.\overline{3} = \frac{1}{3} = C_{\mathrm{cyc}}(W_3^{\mathrm{cyc}})$.

For operation $\circlearrowleft$, the situation is a bit more involved. In general, the property does not hold for iteration. Consider a workflow net $M_n \in \mathcal{M}$ with n nodes and a maximal portion of nodes lying on a cycle. This portion is $\frac{n-2}{n}$, because the start and end place cannot lie on a cycle by definition. Now fix another workflow net $M \in \mathcal{M}$ with k nodes. We show $C_{\mathrm{cyc}}(M_n) > C_{\mathrm{cyc}}(M_n \circlearrowleft M)$ if n is large enough. In $M_n \circlearrowleft M$ exactly 4 nodes that do not lie on a cycle. That means:

$$
\begin{aligned}
& C_{\mathrm{cyc}}(M_n) > C_{\mathrm{cyc}}(M_n \circlearrowleft M) \\
\Leftrightarrow\ & \frac{n-2}{n} > \frac{n+k+6}{n+k+10} \\
\Leftrightarrow\ & (n-2)(n+k+10) > n(n+k+6) \\
\Leftrightarrow\ & n^2 + kn + 10n - 2n - 2k - 20 > n^2 + kn + 6n \\
\Leftrightarrow\ & 8n - 2k - 20 > 6n \\
\Leftrightarrow\ & 2n > 20 + 2k \\
\Leftrightarrow\ & n > 10 + k.
\end{aligned}
$$

Note that such a net M_n is not a meaningful workflow net: Let t be a transition with $p_i \in {}^\bullet t$. Since t lies on a cycle, there is another place $p \in {}^\bullet t$ which is unmarked. Therefore, t is dead and there is no firing sequence from the start to the end place.

If we only consider sound workflow nets, then the property holds for iteration. It is easy to see, that $C_{\mathrm{cyc}}(M) \leq \frac{n-4}{n}$ if M is a sound workflow net with n nodes. We deduce for an arbitrary additional workflow net $M' \in \mathcal{M}$ with k

nodes:

$$C_{\text{cyc}}(M) \leq C_{\text{cyc}}(M \circlearrowleft M')$$
$$\Leftarrow \frac{n-4}{n} \leq \frac{n+k+6}{n+k+10}$$
$$\Leftrightarrow (n-4)(n+k+10) > n(n+k+6)$$
$$\Leftrightarrow n^2 + kn + 10n - 4n - 4k - 40 \leq n^2 + kn + 6n$$
$$\Leftrightarrow 6n - 4k - 40 \leq 6n$$
$$\Leftrightarrow 0 \leq 40 + 4k$$

The last inequation is obviously true.

COMP ($\checkmark$): Take the workflow nets $W_1^{\text{cyc}}, W_2^{\text{cyc}}$ and W_3^{cyc} of Figure 51. We have $C_{\text{cyc}}(W_1^{\text{cyc}}) = 0 = C_{\text{cyc}}(W_2^{\text{cyc}})$, but:

 - $C_{\text{cyc}}(W_1^{\text{cyc}} \rightarrow W_3^{\text{cyc}}) = \frac{2}{10} = 0.2 \neq 0.1\overline{6} = \frac{2}{12} = C_{\text{cyc}}(W_2^{\text{cyc}} \rightarrow W_3^{\text{cyc}})$,
 - $C_{\text{cyc}}(W_1^{\text{cyc}} \wedge W_3^{\text{cyc}}) = \frac{2}{13} \approx 0.1538 \neq 0.1\overline{3} = \frac{2}{15} = C_{\text{cyc}}(W_2^{\text{cyc}} \wedge W_3^{\text{cyc}})$,
 - $C_{\text{cyc}}(W_1^{\text{cyc}} \times W_3^{\text{cyc}}) = \frac{2}{15} = 0.1\overline{3} \neq 0.1176 \approx \frac{2}{17} = C_{\text{cyc}}(W_2^{\text{cyc}} \times W_3^{\text{cyc}})$,
 - $C_{\text{cyc}}(W_1^{\text{cyc}} \circlearrowleft W_3^{\text{cyc}}) = \frac{15}{19} \approx 0.789 \neq 0.810 \approx \frac{17}{21} = C_{\text{cyc}}(W_2^{\text{cyc}} \circlearrowleft W_3^{\text{cyc}})$.

PERM ($\checkmark$): Figure 52 shows two workflow nets, W_4^{cyc} and W_5^{cyc}, that are permutations of each other, but $C_{\text{cyc}}(W_4^{\text{cyc}}) = \frac{1}{3} \neq 0 = C_{\text{cyc}}(W_5^{\text{cyc}})$.
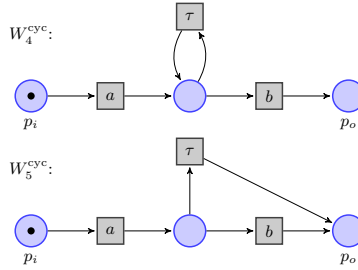

Fig. 52: Two workflow nets with $W_4^{\text{cyc}} \in Perm(W_5^{\text{cyc}})$ and complexity scores $C_{\text{cyc}}(W_4^{\text{cyc}}) = \frac{1}{3} \neq 0 = C_{\text{cyc}}(W_5^{\text{cyc}})$.

ROB$_\ell$ ($\checkmark$): A relabelling of the transitions does not affect C_{cyc}.

¬SUB ($\checkmark$): This property does not hold for operations $\oplus \in \{\rightarrow, \wedge, \times\}$, as shown by the following Theorem:

Theorem 8. *Let $M_1, M_2 \in \mathcal{M}$ be workflow nets and $\oplus \in \{\rightarrow, \wedge, \times\}$ one of the operations of Defintiion 6. Then, $C_{cyc}(M_1 \oplus M_2) < C_{cyc}(M_1) + C_{cyc}(M_2)$.*

Proof. Let $x_\oplus$ be the number of new nodes the operation $\oplus$ introduces in the net $M := M_1 \oplus M_2$. Further, let c_1 be the number of nodes on a cycle in M_1 and c_2 be the number of nodes on a cycle in M_2. Let n_1 be the total number of nodes in M_1 and n_2 be the total number of nodes in M_2. Since

$\oplus$ doesn't introduce a new cycle, all newly introduced nodes lie outside of a cycle and we have

$$C_{\mathrm{cyc}}(M) = \frac{c_1 + c_2}{n_1 + n_2 + x_\oplus} < \frac{c_1 + c_2}{n_1 + n_2} < \frac{c_1}{n_1} + \frac{c_2}{n_2} = C_{\mathrm{cyc}}(M_1) + C_{\mathrm{cyc}}(M_2)$$

$\square$

For the operation $\circlearrowleft$, however, we can find a counter-example for subadditivity: Consider the net W_0 of Figure 2. If we use this net as input for $\circlearrowleft$, we get $C_{\mathrm{cyc}}(W_0 \circlearrowleft W_0) = \frac{12}{16} = 0.75 > 0 + 0 = C_{\mathrm{cyc}}(W_0) + C_{\mathrm{cyc}}(W_0)$.

DEF ($\checkmark$): The smallest possible worfklow net, W_0, has 3 nodes, so the denominator of C_{cyc} is always positive. Furthermore, the number of nodes on a cycle cannot be negative so $C_{\mathrm{cyc}}(M) \geq 0$ for all workflow nets $M \in \mathcal{M}$.

MIN ($\checkmark$): The minimum possible number of nodes lying on a cycle is 0, so 0 is the minimum possible value for C_{cyc}. The workflow net W_0 of Figure 2 receives this complexity score.

INF ($\checkmark$): Let $n \in \mathbb{N}$ with $n \geq 2$. Figure 53 shows how to construct a workflow net with complexity score $\frac{1}{n+1}$ according to C_{cyc}. Therefore, we get that $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{cyc}}(M) = c\}| \geq |\{\frac{1}{n+1} \mid n \in \mathbb{N} \wedge n \geq 2\}| = \infty$.
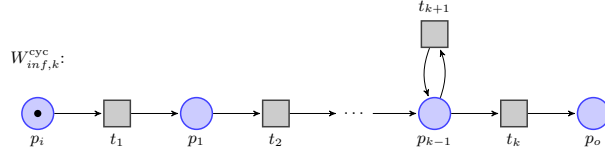

Fig. 53: A workflow net $W_{inf,k}^{\mathrm{cyc}}$ with $k + 1$ transitions and $k + 1$ places. One transition and one place lie on a cycle in thsi net. Its complexity is therefore $C_{\mathrm{cyc}}(W_{inf,k}^{\mathrm{cyc}}) = \frac{2}{2(k+1)} = \frac{1}{k+1}$.

¬SUP ($\checkmark$): Theorem 8 directly implies that C_{cyc} is not superadditive for the operations $\rightarrow, \wedge, \times$. For the operation $\circlearrowleft$, consider the net $W_{\approx 1,k}^{\mathrm{cyc}}$ of Figure 54. For $k \geq 2$, $C_{\mathrm{cyc}}(W_{\approx 1,k}^{\mathrm{cyc}} \circlearrowleft W_{\approx 1,k}^{\mathrm{cyc}}) < 1 < \frac{2k}{k+1} = C_{\mathrm{cyc}}(W_{\approx 1,k}^{\mathrm{cyc}}) + C_{\mathrm{cyc}}(W_{\approx 1,k}^{\mathrm{cyc}})$, so C_{cyc} is also not superadditive for the operator $\circlearrowleft$.

ADD ($\times$): Our analyses of the property ¬SUB show that C_{cyc} is not additive with respect to any of the operations.
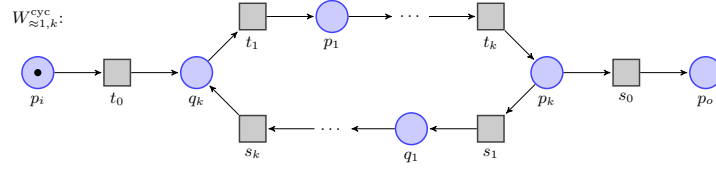
Fig. 54: A workflow net $W_{\approx 1,k}^{\mathrm{cyc}}$ where $4k$ nodes lie on a cycle. In total, there are $4k + 4$ nodes, so $C_{\mathrm{cyc}}(W_{\approx 1,k}^{\mathrm{cyc}}) = \frac{4k}{4k+4} = \frac{k}{k+1}$.

5.4   Degree of Connectedness.

The fourth complexity dimension found by Lieben et al. [9] is labeled DEGREE OF CONNECTEDNESS. Complexity measures of this dimension capture how many sequence flows are possible in the model, since each sequence flow introduces complexity. This dimension contains only two complexity measures, which are quite similar in their behavior.

Coefficient of Network Connectivity The coefficient of network connectivity [3, p120] relates the number of edges in the model to its total number of vertices. The idea is that workflow nets with a low ratio of edges to vertices are more sparse and therefore easier to understand. The advantage of this metric in contrast to density is that it can be adapted to any type of graph, since we don't need to know the maximum possible number of edges in the graph.

$$C_{\mathrm{CNC}}(N) = \frac{|F|}{|P| + |T|} \tag{21}$$

Figure 55 shows example nets and their respective complexity scores.
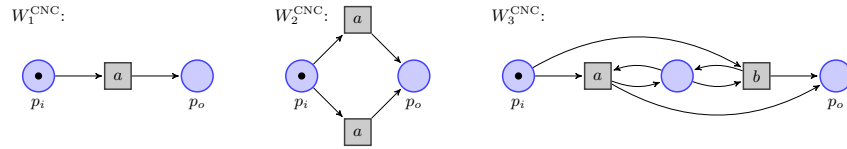
Fig. 55: Three workflow nets, W_1^{CNC}, W_2^{CNC}, W_3^{CNC}, with $C_{\mathrm{CNC}}(W_1^{\mathrm{CNC}}) = \frac{2}{3}$, $C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}}) = 1$ and $C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}}) = \frac{8}{5}$.

¬**TRIV** ($\checkmark$): For the two nets W_1^{CNC} and W_2^{CNC} of Figure 55, we get the complexity scores $C_{\mathrm{CNC}}(W_1^{\mathrm{CNC}}) = \frac{2}{3} \neq 1 = C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}})$.

$W_{1,k}^{\mathrm{CNC}}$:

Fig. 56: A construction plan for workflow nets with $4k+4$ arcs, $2k+2$ places and $2k+2$ transitions. The complexity score of this net is $C_{\mathrm{CNC}}(W_{1,k}^{\mathrm{CNC}}) = \frac{4k+4}{4k+4} = 1$.
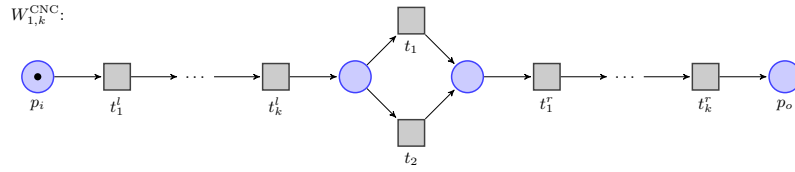
FIN ($\times$): Figure 56 shows how to construct infinitely many workflow nets with complexity 1 according to C_{CNC}, so C_{CNC} does not fulfill this property.

COL ($\checkmark$): Figure 56 shows that there are at least two workflow nets with complexity 1 according to C_{CNC}.

IND$_L$ ($\checkmark$): Take the the workflow nets W_1^{CNC} and W_2^{CNC} of Figure 55. Their languages are $L(W_1^{\mathrm{CNC}}) = \{\varepsilon, a\} = L(W_2^{\mathrm{CNC}})$, but their complexity scores are $C_{\mathrm{CNC}}(W_1^{\mathrm{CNC}}) = \frac{2}{3} \neq 1 = C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}})$.

MON ($\times$): Take the workflow nets W_1^{CNC} and W_3^{CNC} of Figure 55. We get:
- $C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}} \to W_1^{\mathrm{CNC}}) = \frac{12}{5+4} = \frac{12}{9} = 1.\overline{3} < 1.6 = \frac{8}{5} = C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}})$,
- $C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}} \wedge W_1^{\mathrm{CNC}}) = \frac{16}{7+5} = \frac{16}{12} = 1.\overline{3} < 1.6 = \frac{8}{5} = C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}})$,
- $C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}} \times W_1^{\mathrm{CNC}}) = \frac{18}{7+7} = \frac{9}{7} \approx 1.29 < 1.6 = \frac{8}{5} = C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}})$,
- $C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}} \circlearrowleft W_1^{\mathrm{CNC}}) = \frac{22}{9+9} = \frac{11}{9} = 1.\overline{2} < 1.6 = \frac{8}{5} = C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}})$.

COMP ($\checkmark$): Take the workflow nets W_2^{CNC} and W_3^{CNC} of Figure 55, as well as the workflow net $W_{1,1}^{\mathrm{CNC}}$ of Figure 56. The workflow nets W_2^{CNC} and $W_{1,1}^{\mathrm{CNC}}$ have the same complexity score $C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}}) = C_{\mathrm{CNC}}(W_{1,1}^{\mathrm{CNC}}) = 1$. Combining them with the net W_3^{CNC}, however, we get:
- $C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}} \to W_3^{\mathrm{CNC}}) = \frac{14}{5+5} = 1.4$
$\neq 1.286 \approx \frac{18}{7+7} = C_{\mathrm{CNC}}(W_{1,1}^{\mathrm{CNC}} \to W_3^{\mathrm{CNC}})$,
- $C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}} \wedge W_3^{\mathrm{CNC}}) = \frac{18}{7+6} \approx 1.385$
$\neq 1.294 \approx \frac{22}{9+8} = C_{\mathrm{CNC}}(W_{1,1}^{\mathrm{CNC}} \wedge W_3^{\mathrm{CNC}})$,
- $C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}} \times W_3^{\mathrm{CNC}}) = \frac{20}{7+8} = 1.\overline{3}$
$\neq 1.263 \approx \frac{24}{9+10} = C_{\mathrm{CNC}}(W_{1,1}^{\mathrm{CNC}} \times W_3^{\mathrm{CNC}})$,
- $C_{\mathrm{CNC}}(W_2^{\mathrm{CNC}} \circlearrowleft W_3^{\mathrm{CNC}}) = \frac{24}{9+10} \approx 1.263$
$\neq 1.217 \approx \frac{28}{11+12} = C_{\mathrm{CNC}}(W_{1,1}^{\mathrm{CNC}} \circlearrowleft W_3^{\mathrm{CNC}})$.

PERM ($\checkmark$): The workflow net W_3^{CNC} of Figure 55 is a permutation of the net $W_{<1,2}^{\mathrm{CNC}}$ of Figure 57, but $C_{\mathrm{CNC}}(W_3^{\mathrm{CNC}}) = \frac{8}{5} \neq \frac{4}{5} = C_{\mathrm{CNC}}(W_{<1,2}^{\mathrm{CNC}})$.

ROB$_\ell$ ($\checkmark$): C_{CNC} is only dependent on the number of arcs, places and transitions, so performing a relabeling on the transitions cannot have any impact on the complexity score of C_{CNC}.

¬SUB ($\times$): Since we know how many nodes and edges we add with each operation, we can argue over general graphs instead of Petri nets to show that C_{CNC} is subadditive.

Theorem 9. *Let G_1 be a graph with n_1 nodes and $m_1 \geq n_1 - 1$ edges, G_2 be a graph with n_2 nodes and $m_2 \geq n_2 - 1$ edges, and G be a graph with $n = n_1 + n_2 + 1$ nodes and $m = m_1 + m_2 + 2$ edges. Then, $\frac{m_1}{n_1} + \frac{m_2}{n_2} > \frac{m}{n}$.*

Proof.

$$\frac{m_1}{n_1} + \frac{m_2}{n_2} > \frac{m_1 + m_2 + 2}{n_1 + n_2 + 1}$$
$$\Leftrightarrow (m_1 n_1 + m_2 n_1) \cdot (n_1 + n_2 + 1) > (m_1 + m_2 + 2) \cdot (n_1 n_2)$$
$$\Leftrightarrow \underbrace{m_1 n_2^2}_{>n_2} + \underbrace{m_2 n_1^2}_{>n_1} + \underbrace{m_1}_{\geq n_1 - 1} n_2 + \underbrace{m_2}_{\geq n_2 - 1} n_1 > 2 n_1 n_2$$

$\square$

Theorem 9 directly implies that C_{CNC} is subadditive for $\rightarrow$.

Theorem 10. *Let G_1 be a graph with $n_1 \geq 3$ nodes and $m_1 \geq n_1 - 1$ edges, G_2 be a graph with $n_2 \geq 3$ nodes and $m_2 \geq n_2 - 1$ edges, and G be a graph with $n = n_1 + n_2 + k$ nodes and $m = m_1 + m_2 + k + 2$ edges for $k \in \{4, 6, 10\}$. Then, $\frac{m_1}{n_1} + \frac{m_2}{n_2} > \frac{m}{n}$.*

Proof.

$$\frac{m_1}{n_1} + \frac{m_2}{n_2} > \frac{m_1 + m_2 + k + 2}{n_1 + n_2 + k}$$
$$\Leftrightarrow \underbrace{m_1 n_2^2}_{\geq 6 n_2 (*)} + k \underbrace{m_1}_{\geq n_1 - 1} n_2 + \underbrace{m_2 n_1^2}_{\geq 6 n_1 (*)} + k \underbrace{m_1}_{\geq n_2 - 1} n_1 > (k+2) n_1 n_2$$

Where $(*)$ is true because $n_1, n_2 \geq 3$ and $m_1, m_2 \geq 2$. The lower inequality is true since

$$k(n_1 - 1)n_2 + 6n_1 + k(n_2 - 1)n_1 + 6n_2$$
$$= 2k n_1 n_2 - (k - 6)n_1 - (k - 6)n_2$$
$$> (2k - 2(k - 6))n_1 n_2$$
$$= 12 n_1 n_2$$
$$\overset{k \leq 10}{\geq} (k + 2)n_1 n_2$$

$\square$

DEF ($\checkmark$): The coefficient of network connectivity is defined for all workflow nets, since by definition $|P| \geq 2$ and $|T| \geq 1$ in every workflow net. Furthermore, $|F| \geq 0$, so the measure returns only non-negative values.

MIN ($\checkmark$): In a workflow net, every node needs to lie on a path from the input place to the output place, so $|F| \geq |P| + |T| - 1$. Since $\frac{n-1}{n}$ is strictly increasing with increasing n, the smallest possible value of C_{CNC} is $\frac{2}{3}$. This smallest possible complexity score is rewarded to W_0 of Figure 2.
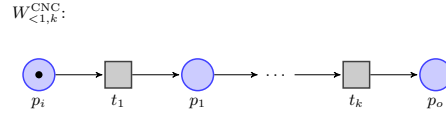
$W_{<1,k}^{\mathrm{CNC}}:$

Fig. 57: A construction plan for workflow nets with $2k$ arcs, $k+1$ places and k transitions. The complexity score of this net is $C_{\mathrm{CNC}}(W_{<1,k}^{\mathrm{CNC}}) = \frac{2k}{2k+1}$.

INF ($\checkmark$): Figure 57 shows how to construct infinitely many workflow nets with different complexity scores. Thus, C_{CNC} doesn't fulfill this property, since $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{CNC}}(M) = c\}| \geq |\{\frac{2k}{2k+1} \mid k \in \mathbb{N}\}| = |\mathbb{N}| = \infty$.

¬SUP ($\checkmark$): The analysis of the property MON shows that C_{CNC} is not super-additive.

ADD ($\times$): The analysis of the property MON shows that C_{CNC} is not additive.

Density The density [3, p.120] of a process model is the number of its arcs divided by the total number of possible arcs. Since a workflow net is a bipartite graph, the maximum possible number of arcs is $|P| \cdot |T| - 2 \cdot |T|$. We therefore define this complexity measure as follows:

$$C_{\mathrm{dens}}(N) = \frac{|F|}{2 \cdot |P| \cdot |T| - 2 \cdot |T|} = \frac{|F|}{2 \cdot |T| \cdot (|P| - 1)}. \tag{22}$$

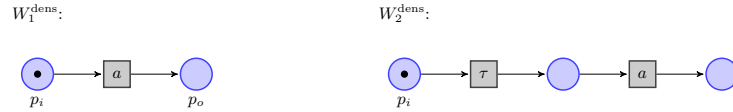Figure 58 shows two simple example nets and their respective complexity scores.

$W_1^{\mathrm{dens}}:$ $W_2^{\mathrm{dens}}:$

Fig. 58: Two workflow nets, W_1^{dens} and W_2^{dens}, with $C_{\mathrm{dens}}(W_1^{\mathrm{dens}}) = 1$ and $C_{\mathrm{dens}}(W_2^{\mathrm{dens}}) = 0.5$.

¬TRIV ($\checkmark$): For the two nets W_1^{dens} and W_2^{dens} of Figure 58, we get the complexity scores $C_{\mathrm{dens}}(W_1^{\mathrm{dens}}) = 1 \neq 0.5 = C_{\mathrm{dens}}(W_2^{\mathrm{dens}})$.

FIN ($\times$): Figure 59 shows how to construct infinitely many workflow nets with complexity 1, so C_{dens} does not fulfill this property.

COL ($\checkmark$): Figure 59 shows that two structurally different workflow nets can get the same complexity score from C_{dens}.

IND$_L$ ($\checkmark$): Take the workflow nets W_1^{dens} and W_2^{dens} of Figure 58. Their languages are $L(W_1^{\mathrm{dens}}) = \{\varepsilon, a\} = L(W_2^{\mathrm{dens}})$, but their complexity scores are $C_{\mathrm{dens}}(W_1^{\mathrm{dens}}) = 1 \neq 0.5 = C_{\mathrm{dens}}(W_2^{\mathrm{dens}})$.
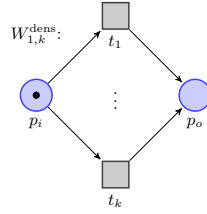
Fig. 59: A workflow net $W_{1,k}^{\mathrm{dens}}$ with 2 places, k transitions and $2k$ edges. For this net and some $k \geq 1$, we get $C_{\mathrm{dens}}(W_{1,k}^{\mathrm{dens}}) = \frac{2k}{2 \cdot k \cdot (2-1)} = 1$.

MON ($\times$): Consider the net W_1^{dens} of Figure 58. We get:

- $C_{\mathrm{dens}}(W_1^{\mathrm{dens}} \to W_1^{\mathrm{dens}}) = \frac{6}{2 \cdot 3 \cdot (4-1)} = \frac{1}{3} < 1 = C_{\mathrm{dens}}(W_1^{\mathrm{dens}})$,
- $C_{\mathrm{dens}}(W_1^{\mathrm{dens}} \wedge W_1^{\mathrm{dens}}) = \frac{10}{2 \cdot 4 \cdot (6-1)} = \frac{1}{4} < 1 = C_{\mathrm{dens}}(W_1^{\mathrm{dens}})$,
- $C_{\mathrm{dens}}(W_1^{\mathrm{dens}} \times W_1^{\mathrm{dens}}) = \frac{12}{2 \cdot 6 \cdot (6-1)} = \frac{1}{5} < 1 = C_{\mathrm{dens}}(W_1^{\mathrm{dens}})$,
- $C_{\mathrm{dens}}(W_1^{\mathrm{dens}} \circlearrowleft W_1^{\mathrm{dens}}) = \frac{16}{2 \cdot 8 \cdot (8-1)} = \frac{1}{7} < 1 = C_{\mathrm{dens}}(W_1^{\mathrm{dens}})$.

Therefore, C_{dens} does not fulfill this property for any of these operations.

COMP ($\checkmark$): Consider the nets W_1^{dens} and W_2^{dens} of Figure 58 as well as the net W_3^{dens} of Figure 60. We get:

- $C_{\mathrm{dens}}(W_2^{\mathrm{dens}} \to W_1^{\mathrm{dens}}) = \frac{8}{2 \cdot 4 \cdot (5-1)} = \frac{1}{4}$
$$\neq \frac{13}{50} = \frac{13}{2 \cdot 5 \cdot (6-1)} = C_{\mathrm{dens}}(W_3^{\mathrm{dens}} \to W_1^{\mathrm{dens}}),$$
- $C_{\mathrm{dens}}(W_2^{\mathrm{dens}} \wedge W_1^{\mathrm{dens}}) = \frac{12}{2 \cdot 5 \cdot (7-1)} = \frac{1}{5}$
$$\neq \frac{17}{84} = \frac{17}{2 \cdot 6 \cdot (8-1)} = C_{\mathrm{dens}}(W_3^{\mathrm{dens}} \wedge W_1^{\mathrm{dens}}),$$
- $C_{\mathrm{dens}}(W_2^{\mathrm{dens}} \times W_1^{\mathrm{dens}}) = \frac{14}{2 \cdot 7 \cdot (7-1)} = \frac{1}{6}$
$$\neq \frac{19}{112} = \frac{19}{2 \cdot 8 \cdot (8-1)} = C_{\mathrm{dens}}(W_3^{\mathrm{dens}} \times W_1^{\mathrm{dens}}),$$
- $C_{\mathrm{dens}}(W_2^{\mathrm{dens}} \circlearrowleft W_1^{\mathrm{dens}}) = \frac{18}{2 \cdot 9 \cdot (9-1)} = \frac{1}{8}$
$$\neq \frac{23}{180} = \frac{23}{2 \cdot 10 \cdot (10-1)} = C_{\mathrm{dens}}(W_3^{\mathrm{dens}} \circlearrowleft W_1^{\mathrm{dens}}).$$

Therefore, C_{dens} fulfills this property for all of these operations.
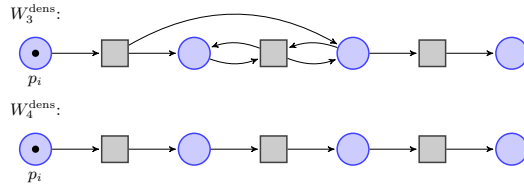
Fig. 60: Two workflow nets, W_3^{dens} and W_4^{dens}, where $W_3^{\mathrm{dens}} \in Perm(W_4^{\mathrm{dens}})$ and $C_{\mathrm{dens}}(W_3^{\mathrm{dens}}) = \frac{9}{2 \cdot 3 \cdot (4-1)} = 0.5$ and $C_{\mathrm{dens}}(W_4^{\mathrm{dens}}) = \frac{6}{2 \cdot 3 \cdot (4-1)} = \frac{1}{3}$.

PERM ($\checkmark$): Figure 60 shows two workflow nets, W_3^{dens} and W_4^{dens}, that are permutations of each other, but $C_{\text{dens}}(W_3^{\text{dens}}) = 0.5 \neq 0.\overline{3} = C_{\text{dens}}(W_4^{\text{dens}})$.

ROB$_\ell$ ($\checkmark$): C_{dens} depends only on the number of arcs, places and transitions. Therefore, the labeling has no impact on the values returned by C_{dens}.

¬SUB ($\times$): Let $M_1, M_2 \in \mathcal{M}$. For $i \in \{1, 2\}$, let f_i be the number of arcs in M_i, t_i be the number of transitions in M_i and p_i be the number of places in M_i. By arithmetic rules, we have:

$$C_{\text{dens}}(M_1) + C_{\text{dens}}(M_2) = \frac{f_1}{2t_1 p_1 - 2t_1} + \frac{f_2}{2t_2 p_2 - 2t_2}$$
$$> \frac{f_1 + f_2}{2t_1 p_1 - 2t_1 + 2t_2 p_2 - 2t_2} =: \frac{n}{d}$$

– The operator $\rightarrow$ introduces 1 transition and 2 arcs. Therefore,

$$C_{\text{dens}}(M_1 \rightarrow M_2) = \frac{n+2}{d + 2t_1 p_2 + 2t_2 p_1 + 2p_1 + 2p_2 - 2} < \frac{n}{d}$$

since $2t_1 p_2 + 2t_2 p_1 + 2p_1 + 2p_2 - 2 > 2$ ($p_i \geq 2$ and $t_i \geq 1$ for all $i \in \{1, 2\}$ by definition of workflow nets).

– The operator $\wedge$ introduces 2 places, 2 transitions and 6 arcs. Therefore,

$$C_{\text{dens}}(M_1 \wedge M_2) = \frac{n+6}{d + 2t_1 p_2 + 4t_1 + 2t_2 p_1 + 4t_2 + 4p_1 + 4p_2 + 4} < \frac{n}{d}$$

since $2t_1 p_2 + 4t_1 + 2t_2 p_1 + 4t_2 + 4p_1 + 4p_2 + 4 > 6$.

– The operator $\times$ introduces 2 places, 4 transitions and 8 arcs. Therefore,

$$C_{\text{dens}}(M_1 \times M_2) = \frac{n+8}{d + 2t_1 p_2 + 6t_1 + 2t_2 p_1 + 6t_2 + 8p_1 + 8p_2 + 16} < \frac{n}{d}$$

since $2t_1 p_2 + 6t_1 + 2t_2 p_1 + 6t_2 + 8p_1 + 8p_2 + 16 > 8$.

– The operator $\circlearrowleft$ introduces 4 places, 6 transitions and 12 arcs. Therefore,

$$C_{\text{dens}}(M_1 \circlearrowleft M_2) = \frac{n+12}{d + 2t_1 p_2 + 8t_1 + 2t_2 p_1 + 8t_2 + 12p_1 + 12p_2 + 48}$$

which is $< \frac{n}{d}$, since $2t_1 p_2 + 8t_1 + 2t_2 p_1 + 8t_2 + 12p_1 + 12p_2 + 48 > 12$.
Thus, $C_{\text{dens}}(M_1 \oplus M_2) < C_{\text{dens}}(M_1) + C_{\text{dens}}(M_2)$ for all $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$ and C_{dens} is subadditive for any of these operations.

DEF ($\checkmark$): The density of a workflow net is always defined, since thre are at least two places and one transition in a workflow net. We therefore cannot divide by zero while evaluating the complexity score. Furthermore, since $|F| \geq 0$ for every workflow net, the values of the complexity metric are non-negative.

MIN ($\times$): C_{dens} has no minimum value that can be reached by a workflow net: Consider the net $W_{\text{min},k}^{\text{dens}}$ of Figure 61. This workflow net has a complexity of $C_{\text{dens}}(W_{\text{min},k}^{\text{dens}}) = \frac{1}{k}$. Therefore, for any workflow net M with complexity $c = C_{\text{dens}}(M) > 0$ we can find a workflow net with smaller complexity score by setting $k > \frac{1}{c}$. A workflow net with complexity 0 cannot exist, since all workflow nets contain at least one arc.
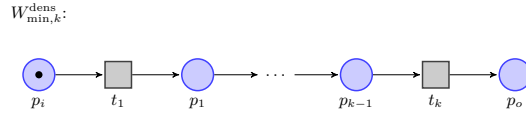
$W_{\text{min},k}^{\text{dens}}$:

Fig. 61: A workflow net $W_{\text{min},k}^{\text{dens}}$ with $k+1$ places, k transitions and $2k$ edges. For this net and some $k \geq 1$, we get $C_{\text{dens}}(W_{\text{min},k}^{\text{dens}}) = \frac{2k}{2 \cdot k \cdot (k+1-1)} = \frac{1}{k}$.

INF ($\checkmark$): Figure 61 shows the construction of infinitely many workflow nets with pairwise distinct complexity according to C_{dens}. With this construction, we get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\text{dens}}(M) = c\}| \geq |\mathbb{N}| = \infty$.

¬SUP ($\checkmark$): The analysis of property MON shows that C_{dens} is not superadditive.

ADD ($\times$): The analysis of property MON shows that C_{dens} is not additive.

5.5 Other Complexity Measures

Lieben et al. [9] highlight that there are two complexity measures that don't fit into any of their discovered dimensions. These complexity measures are the number of empty sequence flows and the number of duplicate tasks. The first evaluates how often a concurrent connector was used in the model that doesn't model concurrent behavior. The latter counts how often a transition label is reused in the model.

Number of Duplicate Tasks The possibility to repeat transition labels often facilitates finding a compact model for a language. However, duplicate labels can also deteriorate the understandability of a process model. If activity labels are repeated throughout the net, it is difficult to see in which cases the activity can occur. Therefore, the amount of label repetitions in a process model can be used as a complexity measure [15]. Let A be an alphabet and $W = (P, T, F, \ell, p_i, p_o)$ be a labeled workflow net with labels from A. For a label $a \in A$, the set of transitions in W with that label is $\ell^{-1}(a) := \{t \in T \mid \ell(t) = a\}$. We define:

$$C_{\text{dup}}(W) = \sum_{a \in A} \left(\max\{|\ell^{-1}(a)|, 1\} - 1 \right) \tag{23}$$

Figure 62 shows example nets and their respective complexity scores.

¬TRIV ($\checkmark$): For the two workflow nets W_1^{dup} and W_2^{dup} of Figure 62, we get the complexity scores $C_{\text{dup}}(W_1^{\text{dup}}) = 0 \neq 2 = C_{\text{dup}}(W_2^{\text{dup}})$.

FIN ($\checkmark$): C_{dup} gives only non-negative integer values as complexity scores. Let A be a finite set of activities and $c \in \mathbb{N}_0$ be a complexity score. A workflow net with complexity score c has c labels that already appeared in the net. There are $c \cdot |A|$ possibilities, which label these duplicate transitions have.
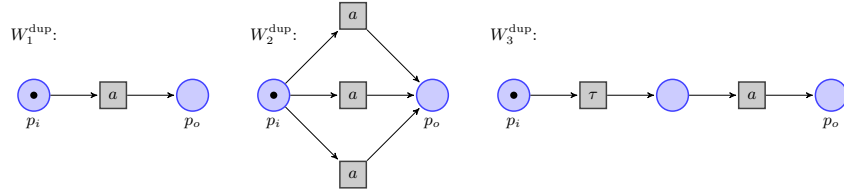
Fig. 62: Three workflow nets, W_1^{dup}, W_2^{dup} and W_3^{dup}, with $C_{\mathrm{dup}}(W_2^{\mathrm{dup}}) = 2$ and $C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) = C_{\mathrm{dup}}(W_3^{\mathrm{dup}}) = 0$.

But in total, the net cannot contain more than $c \cdot |A| + |A| + 1$ transitions, since otherwise there would be more than c duplicates in the net. Therefore, there can only be finitely many workflow nets with complexity c.

COL ($\checkmark$): The workflow nets W_1^{dup} and W_3^{dup} of Figure 62 are different in structure, but both get complexity score $C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) = 0 = C_{\mathrm{dup}}(W_3^{\mathrm{dup}})$.

IND$_L$ ($\checkmark$): Take the workflow nets W_1^{dup} and W_2^{dup} of Figure 62. Their languages are $L(W_1^{\mathrm{dup}}) = L(W_2^{\mathrm{dup}}) = \{\varepsilon, a\}$, but their complexity scores are $C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) = 0 \neq 2 = C_{\mathrm{dup}}(W_2^{\mathrm{dup}})$.

MON ($\checkmark$): C_{dup} is monotone for any operation $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$, since for any $M_1, M_2 \in \mathcal{M}$, a model $M_1 \oplus M_2$ cannot have fewer duplicate labels than M_1 or M_2 have.

COMP ($\checkmark$): Take the workflow net W_1^{dup} of Figure 62. We construct a workflow net $W_{1,1}^{\mathrm{dup}}$ by taking W_1^{dup} and changing the label of its only transition to b. By setting the labels of all transitions that are newly introduced by one of the operations to τ, we get:

- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \rightarrow W_1^{\mathrm{dup}}) = 1 \neq 0 = C_{\mathrm{dup}}(W_{1,1}^{\mathrm{dup}} \rightarrow W_1^{\mathrm{dup}})$,
- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \wedge W_1^{\mathrm{dup}}) = 2 \neq 1 = C_{\mathrm{dup}}(W_{1,1}^{\mathrm{dup}} \wedge W_1^{\mathrm{dup}})$,
- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \times W_1^{\mathrm{dup}}) = 4 \neq 3 = C_{\mathrm{dup}}(W_{1,1}^{\mathrm{dup}} \times W_1^{\mathrm{dup}})$,
- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \circlearrowleft W_1^{\mathrm{dup}}) = 6 \neq 5 = C_{\mathrm{dup}}(W_{1,1}^{\mathrm{dup}} \circlearrowleft W_1^{\mathrm{dup}})$.

PERM ($\times$): By our definition of permutations, we must keep the transitions and their labels in a permutation of a net. This means that the complexity score remains the same for all permutations of a workflow net.

ROB$_\ell$ ($\checkmark$): Uniformly changing the labels of a workflow net can only change the set of duplicate tasks, but not the number of them, since a relabeling is a bijection by our definition. Therefore, C_{dup} fulfills this property.

¬SUB ($\checkmark$): Take the workflow net W_1^{dup} of Figure 62. If we give all transitions that are newly introduced by one of the operations the label τ, we get:

- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \rightarrow W_1^{\mathrm{dup}}) = 1 > 0 + 0 = C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) + C_{\mathrm{dup}}(W_1^{\mathrm{dup}})$,
- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \wedge W_1^{\mathrm{dup}}) = 2 > 0 + 0 = C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) + C_{\mathrm{dup}}(W_1^{\mathrm{dup}})$,
- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \times W_1^{\mathrm{dup}}) = 4 > 0 + 0 = C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) + C_{\mathrm{dup}}(W_1^{\mathrm{dup}})$,
- $C_{\mathrm{dup}}(W_1^{\mathrm{dup}} \circlearrowleft W_1^{\mathrm{dup}}) = 6 > 0 + 0 = C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) + C_{\mathrm{dup}}(W_1^{\mathrm{dup}})$.

DEF ($\checkmark$): This metric is obviously defined for every labeled workflow net. Because we take the maximum of $|\ell^{-1}(a)|$ and 1 for any $a \in A$, every summand of the complexity score is non-negative. Therefore, the total score of this complexity measure is non-negative.

MIN ($\checkmark$): The workflow net W_1^{dup} of Figure 62 gets the smallest possible complexity score $C_{\mathrm{dup}}(W_1^{\mathrm{dup}}) = 0$. Since we sum over non-negative values to calculate the complexity score, it is not possible to get a score less than 0.

INF ($\checkmark$): Let $c \in \mathbb{N}_0$. Figure 63 shows how to construct a workflow net with complexity score c, so $|c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_{\mathrm{dup}}(M) = c\} \geq \mathbb{N}_0 = \infty$.
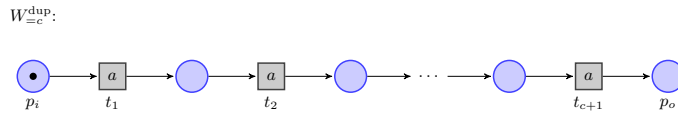
$W_{=c}^{\mathrm{dup}}$:

Fig. 63: A construction plan for workflow nets $W_{=c}^{\mathrm{dup}}$ with $c+1$ transitions and complexity $C_{\mathrm{dup}}(W_{=c}^{\mathrm{dup}}) = c$.

¬SUP ($\times$): Whether C_{dup} fulfills this property depends on the set of activity labels A. Let $M_1, M_2 \in \mathcal{M}$ be two models with distinct transition labels. If A is sufficiently large to give every transition that is newly introduced by $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$ a different label that doesn't occur in M_1 or M_2, we get $C_{\mathrm{dup}}(M_1 \oplus M_2) = C_{\mathrm{dup}}(M_1) + C_{\mathrm{dup}}(M_2)$. On the other hand, if A is not sufficiently large to do so, we add at least one new duplicate label to $M_1 \oplus M_2$. In this case, we get $C_{\mathrm{dup}}(M_1 \oplus M_2) > C_{\mathrm{dup}}(M_1) + C_{\mathrm{dup}}(M_2)$.

ADD ($\times$): The analysis of property ¬SUB shows that C_{dup} is not additive for any of the operations.

Number of Empty Sequence Flows The measure for empty sequence flows is based on work of Gruhn et al. [16] on reducing the complexity of BPMN. They found that, in BPMN models, there are often edes that directly connect and-splits and -joins. These edges can be removed completely, as they don't change the behavior of the net. Regarding workflow nets, there can't be any edges between two and-connectors, since and-connectors are always transitions and Petri nets are bipartite. Instead, empty sequence flows in workflow nets are places that have only and-splits in their preset and and-joins in their postset. We therefore define:

$$C_{\emptyset}(W) = |\{p \in P \mid {}^{\bullet}p \subseteq \mathcal{S}_{\mathrm{and}}^{W} \wedge p^{\bullet} \subseteq \mathcal{J}_{\mathrm{and}}^{W}\}| \tag{24}$$

Figure 64 shows two example nets and their respective complexity scores.

¬TRIV ($\checkmark$): For the two workflow nets $W_1^{\emptyset}$ and $W_2^{\emptyset}$ of Figure 64, we get the complexity scores $C_{\emptyset}(W_1^{\emptyset}) = 0 \neq 2 = C_{\emptyset}(W_2^{\emptyset})$.
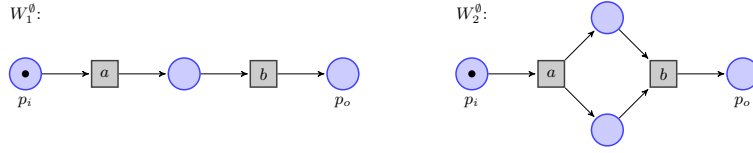
$W_1^{\emptyset}$:

$W_2^{\emptyset}$:

Fig. 64: Two workflow nets, $W_1^{\emptyset}$ and $W_2^{\emptyset}$, with $C_{\emptyset}(W_1^{\emptyset}) = 0$ and $C_{\emptyset}(W_2^{\emptyset}) = 2$.
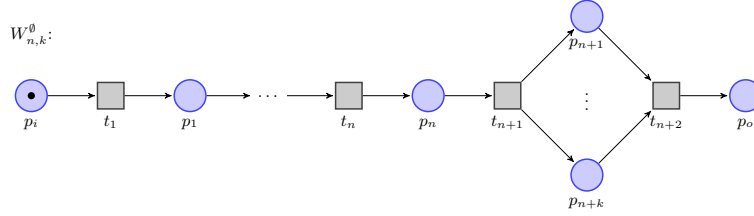
$W_{n,k}^{\emptyset}$:

Fig. 65: A workflow net with $n + k + 2$ places, $n + 2$ transitions and complexity score $C_{\emptyset}(W_{n,k}^{\emptyset}) = k$.

FIN ($\times$): Figure 65 shows how to construct infinitely many workflow nets with the same complexity score $k \in \mathbb{N}$, where $k \geq 2$.

COL ($\checkmark$): Let $k \in \mathbb{N}$ with $k \geq 2$. For any $n \in \mathbb{N}$, $W_{n,k}$ gets the complexity score k, so we have found two workflow nets that receive the same complexity score from $C_{\emptyset}$.

IND$_L$ ($\checkmark$): Take the workflow nets $W_1^{\emptyset}$ and $W_2^{\emptyset}$ of Figure 64. Their languages are $L(W_1^{\emptyset}) = \{\varepsilon, a, ab\} = L(W_2^{\emptyset})$, but their respective complexity scores are $C_{\emptyset}(W_1^{\emptyset}) = 0 \neq 2 = C_{\emptyset}(W_2^{\emptyset})$.

MON ($\checkmark$): None of the operations of Definition 6 can produce empty sequence flows, since a workflow net needs to consist of at least one transition. We therefore get the following Theorem:

Theorem 11. *Let $M_1, M_2 \in \mathcal{M}$ be two workflow nets. Then,*
- $C_{\emptyset}(M_1 \rightarrow M_2) = C_{\emptyset}(M_1) + C_{\emptyset}(M_2),$
- $C_{\emptyset}(M_1 \wedge M_2) = C_{\emptyset}(M_1) + C_{\emptyset}(M_2),$
- $C_{\emptyset}(M_1 \times M_2) = C_{\emptyset}(M_1) + C_{\emptyset}(M_2),$
- $C_{\emptyset}(M_1 \circlearrowleft M_2) = C_{\emptyset}(M_1) + C_{\emptyset}(M_2).$

Proof. Obvious by definition of $C_{\emptyset}$ and the operations $\rightarrow, \wedge, \times, \circlearrowleft$. $\square$

Therefore, for any $M_1, M_2 \in \mathcal{M}$ and any operation $\oplus \in \{\rightarrow, \wedge, \times, \circlearrowleft\}$, we get $C_{\emptyset}(M_1 \oplus M_2) = C_{\emptyset}(M_1) + C_{\emptyset}(M_2) \geq C_{\emptyset}(M_1)$, so $C_{\emptyset}$ fulfills this property.

COMP ($\times$): Let $M_1, M_2, M_3 \in \mathcal{M}$ be three workflow nets with complexity scores $C_{\emptyset}(M_1) = C_{\emptyset}(M_2)$. Theorem 11 gives:
- $C_{\emptyset}(M_1 \rightarrow M_3) = C_{\emptyset}(M_1) + C_{\emptyset}(M_3)$
  $= C_{\emptyset}(M_2) + C_{\emptyset}(M_3) = C_{\emptyset}(M_2 \rightarrow M_3),$

$$- \ C_\emptyset(M_1 \wedge M_3) = C_\emptyset(M_1) + C_\emptyset(M_3)$$
$$= C_\emptyset(M_2) + C_\emptyset(M_3) = C_\emptyset(M_2 \wedge M_3),$$
$$- \ C_\emptyset(M_1 \times M_3) = C_\emptyset(M_1) + C_\emptyset(M_3)$$
$$= C_\emptyset(M_2) + C_\emptyset(M_3) = C_\emptyset(M_2 \times M_3),$$
$$- \ C_\emptyset(M_1 \circlearrowleft M_3) = C_\emptyset(M_1) + C_\emptyset(M_3)$$
$$= C_\emptyset(M_2) + C_\emptyset(M_3) = C_\emptyset(M_2 \circlearrowleft M_3).$$

PERM ($\checkmark$): The workflow nets $W_3^\emptyset$ and $W_4^\emptyset$ of Figure 66 are permutations of each other, but $C_\emptyset(W_3^\emptyset) = 0 \neq 2 = C_\emptyset(W_4^\emptyset)$.
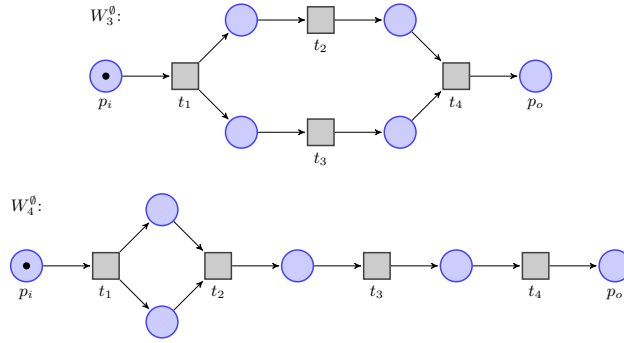

Fig. 66: Two workflow nets, $W_3^\emptyset$ and $W_4^\emptyset$, which are permutations of each other. $C_\emptyset(W_3^\emptyset) = 0$, but $C_\emptyset(W_4^\emptyset) = 2$.

ROB$_\ell$ ($\checkmark$): The number of empty sequence flows is independent of the labeling, so $C_\emptyset$ fulfills this property.

$\neg$SUB ($\times$): Theorem 11 implies that $C_\emptyset$ is subadditive.

DEF ($\checkmark$): $C_\emptyset$ is defined for any workflow net W, since S_{and}^W and J_{and}^W are. It only returns non-negative values, since the smallest possible set has cardinality 0.

MIN ($\checkmark$): The workflow net $W_1^\emptyset$ of Figure 64 has no empty sequence flows, so it gets a complexity score of 0. Because $C_\emptyset$ can only return non-negative values, this is the smallest possible complexity score.

INF ($\checkmark$): Let $k \in \mathbb{N}$ with $k \geq 2$. Figure 65 shows how to construct a workflow net with complexity score k, so $C_\emptyset$ can return infinitely many values and we get $|\{c \in \mathbb{R} \mid \exists M \in \mathcal{M} : C_\emptyset(M) = c\}| \geq |\{k \in \mathbb{N} \mid k \geq 2\}| = \infty$.

$\neg$SUP ($\times$): Theorem 11 implies that $C_\emptyset$ is superadditive.

ADD ($\checkmark$): Theorem 11 implies that $C_\emptyset$ is additive.

6   Discussion of the Results

Table 1 shows the results of our analyses. For properties that depend on which operation of Figure 3 we choose, we added a more detailed analysis in Table 2.

We split the discussion into two parts: We begin by analyzing the properties we used with regard to our findings. This aids people that need to choose complexity measures to interpret our results appropriately. Afterward, we compare the examined complexity measures based on which properties they fulfill.

6.1   Interpreting the Properties

Weyuker proposes that all complexity measures for software programs should fulfill her properties [4]. For complexity measures of process models, we disagree.

Looking at Table 1, our first observation is that some of Weyuker's properties are fulfilled by all measures we analyzed. In particular, this is true for the properties ¬TRIV, COL and IND_L. This comes to no surprise, since useful complexity measures should fulfill these properties: A measure is not reflecting the complexity of a net if it gives every net the same score or finds unique numerical representations of the net or its language. Therefore, we argue that complexity measures for process model complexity should fulfill these three properties.

Another immediate observation is that almost all complexity measures fulfill PERM and ROB_ℓ. The complexity measure C_{dup} is the only exception, and we will see in the second part of the discussion why this is the case. In our opinion, it makes sense for complexity measures to react to permuting the net. Permutations can rearrange the entire structure of a process model, which should be captured by a complexity measure. We therefore argue that PERM is a property all complexity measures should fulfill. However, we don't get to the same conclusion for ROB_ℓ. Even though most measures are robust to relabeling the transitions, the labeling has an undeniable impact on the understandability of a net. Therefore, ROB_ℓ has a descriptive nature and separates measures that consider the labeling from measures that don't.

Property FIN is only fulfilled by two complexity measures. As already argued in Section 4, we think that FIN does not fully convey the intended meaning. Limiting the number of workflow nets that receive the same complexity score is too restrictive. Instead, one should make sure that there aren't only few scores awarded to workflow nets. Our newly introduced property INF examines this and should be fulfilled by every useful complexity measure.

Weyuker's property COMP is designed to test whether complexity measures are sensitive to dependencies between different parts of the net, but needs to be taken carefully in the context of process models. For example, the measure C_{MM} can "repair" existing connector mismatches by introducing even more of them to the net. This is no expected behavior for a complexity measure, but leads to COMP being fulfilled for C_{MM}. Since it is possible to fulfill COMP in this way, we propose to take this property as optional for a complexity measure. We advise taking care if a measure fulfills this property and check whether it does so for the intended reasons.

Table 1: The results of our analyses. Entries with an asterisk imply that the answer depends on the used operation.

	Token Behavior Complexity							Node IO Complexity		
	C_{size}	C_{MM}	C_{CH}	C_{CC}	C_{ts}	C_{sep}	C_{CFC}	C_{mcd}	C_{seq}	C_{acd}
¬TRIV	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FIN	✓	×	×	×	×	×	×	×	×	×
COL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IND_L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MON	✓	×	×	×	✓	×	✓	✓	×	×
COMP	×	✓	✓	✓	×	✓	×	✓	✓	✓
PERM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ROB_ℓ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
¬SUB	✓	×	✓	✓	×*	×*	×*	×	×*	×
DEF	✓	✓	×	✓	✓	✓	✓	×	✓	×
MIN	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
INF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
¬SUP	×	✓	✓	✓	×	✓	×	✓	✓	✓
ADD	×	×	×	×	×*	×	×*	×	×	×

	Path Complexity			Degree of Connectedness		Other	
	C_{depth}	C_{diam}	C_{cyc}	C_{CNC}	C_{dens}	C_{dup}	$C_{\emptyset}$
¬TRIV	✓	✓	✓	✓	✓	✓	✓
FIN	×	×	×	×	×	✓	×
COL	✓	✓	✓	✓	✓	✓	✓
IND_L	✓	✓	✓	✓	✓	✓	✓
MON	✓	✓	×	×	×	✓	✓
COMP	×*	×	✓	✓	✓	✓	×
PERM	✓	✓	✓	✓	✓	×	✓
ROB_ℓ	✓	✓	✓	✓	✓	✓	✓
¬SUB	✓	✓	×*	×	×	✓	×
DEF	✓	✓	✓	✓	✓	✓	✓
MIN	✓	✓	✓	✓	×	✓	✓
INF	✓	✓	✓	✓	✓	✓	✓
¬SUP	✓	×*	✓	✓	✓	×	×
ADD	×	×	×	×	×	×	✓

Table 2: A more detailed overview for measures where it depends on the operation whether a property is fulfilled.

	C_{CFC}	C_{sep}	C_{ts}	C_{seq}	C_{depth}	C_{diam}	C_{cyc}
COMP	×	✓	×	✓	→: ✓ ∧: × ×: × ↻: ×	×	✓
¬SUB	→: × ∧: ✓ ×: ✓ ↻: ✓	→: × ∧: ✓ ×: ✓ ↻: ✓	→: × ∧: ✓ ×: × ↻: ×	→: × ∧: ✓ ×: ✓ ↻: ✓	✓	✓	→: × ∧: × ×: × ↻: ✓
¬SUP	×	✓	×	✓	✓	→: × ∧: ✓ ×: ✓ ↻: ✓	✓
ADD	→: ✓ ∧: × ×: × ↻: ×	×	→: ✓ ∧: × ×: ✓ ↻: ✓	×	×	×	×

The properties MON and ¬SUB are optional for complexity measures. Property MON is very intuitive, but a composed model can always lower the complexity by introducing more structure to a net. Not fulfilling property ¬SUB, on the other hand, might even be advantageous for certain use-cases: If a measure doesn't fulfill ¬SUB, we cannot increase the complexity of two nets beyond the sum of their complexity scores when combining them with one of the operations shown in Figure 3. Our new properties ¬SUP and ADD deepen these insights, but are also intended as optional properties.

Finally, it is dependent on the use-case whether our new properties DEF and MIN are necessary for a complexity measure. Especially since we can define special values for undefined cases, complexity measures that don't fulfill DEF can be as useful as those that do. However, in these cases, we use an altered definition for measuring complexity, and one could argue that this is a different measure than the original one. Regarding this argument, we propose that DEF should be fulfilled by complexity measures. The property MIN is important for applications that iteratively make existing process models less complex, to avoid nondeterminism. For other use-cases, it is sufficient to be aware of whether the complexity measure has a minimum value or not. Therefore, it is optional for a complexity measure to fulfill MIN. An overview over the results of this part of the discussion can be found in Table 3.

Table 3: A table classifying which properties of Section 4 should be fulfilled by a complexity measure (N: normative) or are optional for a complexity measure (D: descriptive).

	WEYUKER'S PROPERTIES								
Property	¬TRIV	FIN	COL	IND_L	MON	COMP	PERM	ROB_ℓ	¬SUB
Classification	N	D	N	N	D	D	N	D	D

	EXTENSIONS				
Property	DEF	MIN	INF	¬SUP	ADD
Classification	N	D	N	D	D

6.2 Comparison of Complexity Measures

It comes to no surprise that the complexity measure based on the size of a workflow net fulfills almost all of Weyuker's properties. Its simplicity and strong connection to understandability [8] further emphasize that the size of a workflow net is an important factor for complexity. However, C_{size} is superadditive, so combining two nets with one of the operations of Figure 3 is guaranteed to return a net with higher complexity than the sum of the input nets. The complexity measures based on connector mismatches and connector heterogeneity are not monotone, since adding certain structures to a workflow net can lower the scores of these complexity measures. If users favor monotonicity, they should avoid these measures and use the control flow complexity measure instead, which also takes the different connectors in a net into account. Furthermore, users of the connector heterogeneity measure should be aware that C_{CH} is not defined for workflow nets that don't contain any connectors. The token split measure is additive for all operations except $\wedge$. Using the operator $\wedge$ to combine n nets, the complexity score of the resulting net is exactly by $n-1$ higher than the sum of the complexities, which can often be tolerated in practice. This measure has the most of what we would call desirable features in the dimension TOKEN BEHAVIOR COMPLEXITY. Separability is not monotone, since all operations of Figure 3, except $\times$, introduce new cut-vertices. The measure is also not monotone with respect to $\times$ if we take MON strictly: If we use a workflow net with complexity 1 as the first argument of $\times$, the resulting net will always fail by definition of MON if it contains at least one cut-vertex. If we weaken MON to avoid this special case, C_{sep} would be monotone for $\times$. Like the control flow complexity measure, C_{sep} is subadditive only for $\rightarrow$. C_{CFC}, however, is also additive for this operation. The measure C_{CC} fulfills COMP for the operator $\rightarrow$ only because the weights of xor-connectors depend on whether they are the input or output place of the net. Furthermore, C_{CC} has no minimum value.

Regarding NODE IO COMPLEXITY, only the complexity measure based on the maximum connector degree is monotone. This measure is also subadditive for workflow nets that have at least one connector. For workflow nets without connectors, however, this measure is undefined and setting the score to 0 in these

cases would destroy subadditivity. The same is true for C_{acd}. Sequentiality C_{seq} is a good choice in this dimension, if ¬SUB needs to be fulfilled.

The measures for diameter and depth of the PATH COMPLEXITY dimension are quite similar, but have different behavior for compositions with the $\rightarrow$ operator. For $\rightarrow$, C_{diam} is superadditive and C_{depth} is sensitive to compositions. Apart from this, these complexity measures have similar properties.

The dimension labeled DEGREE OF CONNECTEDNESS contains no monotone measures. This could hint to a gap for monotone complexity measures that take the connectedness of the net into account. Furthermore, all measures are subadditive, so if this property is undesired, this dimension offers no alternatives. The measures C_{CNC} and C_{dens} are quite similar in their definition and therefore in the properties they fulfill, but C_{dens} offers no minimum complexity score, while C_{CNC} does.

The complexity measure based on duplicate tasks is the only one that doesn't fulfill PERM and ROB$_\ell$, since it solely depends on the labeling of the net. Its superadditivity, however, depends on the number of task labels in the alphabet that aren't already used in the models. The number of empty sequence flows is additive for all operations of Figure 3 and therefore not sensitive to compositions. It is a simple complexity measure, but shows immediate ways to make a model simpler.

6.3   Experiments with an evolutionary discovery algorithm

We conducted a small case study to observe how the properties of Section 4 can drive the evolution of process models in the ETM. We picked three complexity measures with quite different formal properties: C_{size}, C_{acd} and C_{CH}. In each iteration, the ETM chooses a random mutation for a tree in its population. One such mutation can be seen in Figure 67: Process Tree T_1 is mutated into T_2 by adding a $\times$ node as the new root of T_1 and activity a as its children.
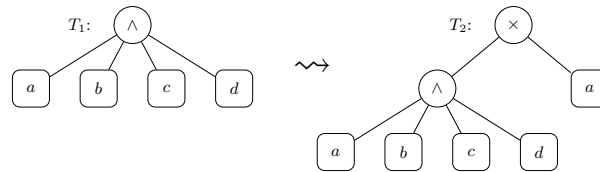

Fig. 67: A possible mutation of a process tree T_1 into T_2.

For C_{size}, this would mean a significant increase in complexity from 16 to 25, while the complexity according to C_{acd} would slightly decrease from 5 to 3.5. According to C_{CH}, the complexity would drop from its highest value 1 to its lowest value 0. This is because C_{acd} is subadditive, while C_{size} is superadditive. C_{CH} is neither sub- nor superadditive, so it prone to high fluctuation. The fact that C_{size} is superadditive can be problematic, since the ETM must

perform several deletion steps to be able to lower the complexity, but it is random whether a deletion is performed during a mutation. Thus, for superadditive complexity measures, ETM needs a higher probability for removing nodes than for subadditive complexity measures to find less complex models.

To confirm this theory, we implemented a simple version of the ETM[3] with a population size of 10 and an elite size of 1. We monitored the evolution of a tree in the population with respect to its complexity, leading to the results shown in Figure 68. This way, we can witness the effect described earlier, as C_{size} gets much higher values and needs to perform several deletion steps to lower the complexity. C_{acd} has a smaller range and a more stable complexity score, but its complexity graph is not continuous, since C_{acd} is not always defined. Finally, C_{CH} fluctuates as strongly as expected.
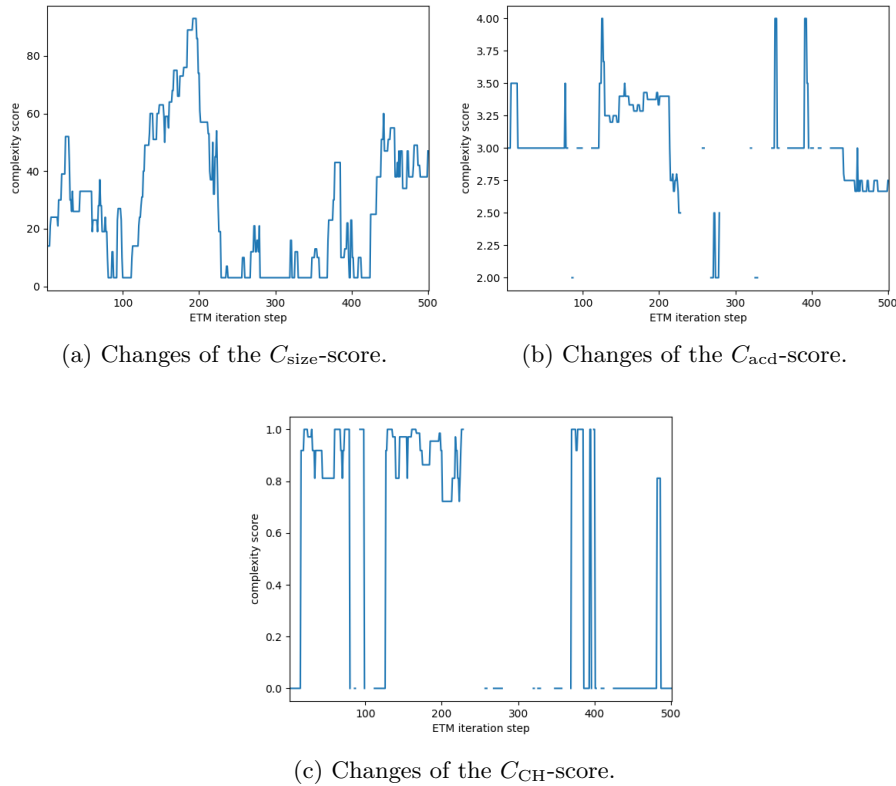
(a) Changes of the C_{size}-score.

(b) Changes of the C_{acd}-score.

(c) Changes of the C_{CH}-score.

Fig. 68: Development of different complexity scores during a run of the Evolutionary Tree Miner.

_____

[3] see https://github.com/Pati-nets/Navigating-Complexity

7   Conclusion

In this paper, we used and extended the properties defined by Weyuker [4] to analyze and compare popular complexity measures for process models [3]. We discussed which of the inspected properties are normative and which are descriptive for process models to give a sense of their importance. Furthermore, we compared complexity measures that are in the same complexity dimension discovered by Lieben et al. [9] and highlighted when to prefer which measure. Defining and analyzing the structuredness measure was out of scope of this paper, so possible future work involves the analysis of this measure. Moreover, one can think of more properties that highlight interesting characteristics of complexity measures. More composition rules than those used by the ETM would be interesting to analyze. Furthermore, investigating how to define permutations of Petri nets is interesting for a more expressive PERM property. The analysis could also be extended to other workflow net complexity measures, such as in [17].

We are confident that our analyses and discussions shed a new light on popular complexity measures that will help analysts and algorithm designers to choose measures fitting to their needs. We also hope to start a discussion on complexity and simplicity measures, which are often overlooked during the evaluation of process models.

References

1. W.M.P. van der Aalst, *"Process mining: Data science in action"* (2016) Berlin, Heidelberg: Springer, https://doi.org/10.1007/978-3-662-49851-4
2. S. Leemans, E. Poppe, M. Wynn, *"Directly follows-based process mining: Exploration and a a case study,"* (2019) ICPM 2019, pp. 25–32, https://doi.org/10.1109/ICPM.2019.00015
3. J. Mendling, *"Metrics for process models"*, (2008) Berlin, Heidelberg: Springer, https://doi.org/10.1007/978-3-540-89224-3
4. E.J. Weyuker *"Evaluating software complexity measures,"* (1988) in IEEE Transactions on Software Engineering, Volume 14, Issue 9, pp. 1357–1365, https://doi.org/10.1109/32.6178
5. J. Cardoso, *"Control-flow complexity measurement of processes and Weyuker's properties,"* (2005) in 6th International Enformatika Conference, Volume 8, pp. 213–218, https://enformatika.org/data/v8/v8-42.pdf
6. W.M.P. van der Aalst, *"Relating process models and event logs - 21 conformance propositions"* (2018) ATAED@Petri Nets/ACSD https://ceur-ws.org/Vol-2115/ATAED2018-56-74.pdf
7. C. Favre, D. Fahland, H. Völzer *"The relationship between workflow graphs and free-choice workflow nets,"* (2015) in Information Systems, Volume 47, pp. 197–219, https://doi.org/10.1016/j.is.2013.12.004
8. H.A. Reijers, J. Mendling, *"A study into the factors that influence the understandability of business process models,"* (2011) in IEEE Transactions on Systems, Man, and Cybernetics, Volume 41, Number 3, pp. 449–462, htpps://doi.org/10.1109/TSMCA.2010.2087017

9. J. Lieben, B. Depaire, M. Jans, T. Jouck, *"An improved way for measuring simplicity during process discovery,"* (2018) in Enterprise and Organizational Modeling and Simulation. LNBIP, vol 332, https://doi.org/10.1007/978-3-030-00787-4_4

10. A.F. Syring, N. Tax, W.M.P. van der Aalst, *"Evaluating Conformance Measures in Process Mining Using Conformance Propositions,"* (2019) in Transactions on Petri Nets and Other Models of Concurrency XIV, LNCS, Volume 11790, https://doi.org/10.1007/978-3-662-60651-3_8

11. J.C.A.M. Buijs, B.F. van Dongen, W.M.P. van der Aalst, *"Quality dimensions in process discovery: the importance of fitness, precision, generalization and simplicity.* (2014) in International Journal of Cooperative Information Systems, 23(1), https://doi.org/10.1142/S0218843014400012

12. S. Leemans, D. Fahland, W.M.P. van der Aalst, *"Discovering block-structured process models from event logs - a constructive approach,"* (2013) in Petri Nets 2013, LNCS, vol. 7927. Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-38697-8_17

13. S. Morasca, *"Measuring attributes of concurrent software specifications in Petri nets,"* (1999), Sixth International Software Metrics Symposium, pp. 100–110, https://doi.org/10.1109/METRIC.1999.809731

14. I. Vanderfeesten, H.A. Reijers, J. Mendling, W.M.P. van der Aalst, J. Cardoso, *"On a quest for good process models: the cross-connectivity metric,"* (2008) in CAiSE 2008, LNCS, vol. 5074, https://doi.org/10.1007/978-3-540-69534-9_36

15. M. La Rosa, P. Wohed, J. Mendling, A.H.M. ter Hofstede, H.A. Reijers and W.M.P. van der Aalst, *"Managing Process Model Complexity Via Abstract Syntax Modifications,"* (2011) in IEEE Transactions on Industrial Informatics, vol. 7, no. 4, pp. 614-629, https://doi.org/10.1109/TII.2011.2166795

16. V. Gruhn, R. Laue, *"Reducing the cognitive complexity of business process models,"* (2009) in 8th IEEE International Conference on Cognitive Informatics, pp. 339-345, https://doi.org/10.1109/COGINF.2009.5250717

17. K.B. Lassen, W.M.P. van der Aalst, *"Complexity metrics for Workflow nets,"* (2009) Information and Software Technology, vol. 51, Issue 3, pp. 610-626, https://doi.org/10.1016/j.infsof.2008.08.005