Pan-European University, Faculty of Informatics

Interim Research Report

Requirements on a low-code language based on object-centric processes

of the project

Requirements and formal definition of a low-code language based on object-centric processes - LowcodeOCP

Authors:
Gabriel Juhás, Milan Mladoniczky, Juraj Mažári and Tomáš Kováčik
Contact:
gabriel.juhas@paneurouni.com, milan.mladoniczky@paneurouni.com

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No.

09I03-03-V04-00493

September 30, 2025
Version: v0.1

Abstract

This interim report summarises the current progress of the research project, outlines objectives and milestones achieved to date, and identifies remaining work, risks, and mitigation plans. Replace this paragraph with your 150–250 word abstract.

Executive Summary

Provide a non-technical 1–2 page summary: objectives, scope, key achievements so far, preliminary results, notable deviations from plan, upcoming milestones, and high-level risks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Context of the problem, why it matters, and who benefits.

1.2 Objectives and Scope

List primary and secondary objectives. Define the scope and out-of-scope items.

1.3 Research Questions / Hypotheses

State the research questions or hypotheses driving the work.

Chapter 2

Related Work

Briefly survey the most relevant literature. Use citations like [**knuth1984texbook**]. Contrast existing approaches and identify gaps this project addresses.

Chapter 3

Methods and Approach

3.1 Methodology

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VAIA VÝSKUMNÁ AGENTÚRA

Chapter 4

Place/transition nets

4.1 Mathematical preliminaries

We use $\mathbb{N}$ to denote the nonnegative integers and $\mathbb{N}^+$ to denote the positive integers. Given two arbitrary sets A and B, the symbol B^A denotes the set of all functions from A to B. Given a function f from A to B and a subset C of A we write $f|_C$ to denote the restriction of f to the set C. The symbol 2^A denotes the power set of a set A. Given a set A, the symbol $|A|$ denotes the cardinality of A and the symbol id_A the identity function on the set A. We write id to denote id_A whenever A is clear from the context. The set of all multisets over a set A is denoted by $\mathbb{N}^A$. The addition of multisets over a finite set A is denoted by $+$. Given two multisets m and m' over A, $m + m'$ is defined by $\forall a \in A : (m + m')(a) = m(a) + m'(a)$. Notice that $(\mathbb{N}^A, +)$ is the free commutative monoid over A. We do not distinguish between a subset $X \subseteq A$ and its characteristic multiset m_X given by $m(x) = 1$ for each $x \in X$ and $m(x') = 0$ for each $x' \in A \setminus X$. Finally, we write as usual $\sum_{a \in A} m(a)a$ to denote the multiset m over A. Given a binary relation $R \subseteq A \times A$ over a set A, the symbol R^+ denotes the transitive closure of R and R^* the reflexive and transitive closure of R.

4.2 Petriflow process definition

Let us now define formal semantics for Petriflow processes.

Definition 1 (Petriflow process)
A Petriflow object-centric process *is a triple* $OCP = (Data, Worfklow, Interface)$, *where:*

- *Data is a triple* $Data = (Variables, Types, Typing)$, *such that:*
  - $Variables$ *is a finite set of data variables*

- $Types$ *is a set of data types such that for each data type* $V \in Types$, V *is a set, called values of the data type* V. *We also consider a unique element null such that for each* V $inTypes$ $null \notin V$.
- $Typing$ *is a function* $Typing : Variables \to Types$ *associating a data type to each data variable.*

- $Workflow$ *is a six-tuple* $Workflow = (P, T, F, C_+, C_-, W)$, *where:*
  - P *is a finite set of* places,
  - T *is a finite set of* tasks, *satisfying* $P \cap T = \emptyset$,
  - $F \subseteq (P \times T) \cup (T \times P)$ *is a* flow relation,
  - $C_+, C_- \subseteq P \times T$ *are positive and negative context relations satisfying* $(F \cup F^{-1}) \cap (C_+ \cup C_-) = C_+ \cap C_- = \emptyset$,
  - $W : F \cup C_+ \cup C_- \to \mathbb{N}^+ \cup P \cup D$ *is a weight function.*

- $Interface$ *is a function* $Interface : T \to 2^{Variables}$ *associating a subset of data variables to each task. Given a task* $t \in T$ *elements of* $Interface(t)$ *are called data fields or data refs of the task* t *and* $Interface(t)$ *itself is also called a form of task* t.

Definition 2 (State of Petriflow OCP) *A state of a Petriflow OCP as defined in Definition 1 is a function* s *with definition domain* $P \cup V \cup T$ *such that:*

- $s|_P : P \to \mathbb{N}$ *is a function that associates a nonnegative marking with each place,*
- *for each* $v \in V : s(d)$ $inTyping(s)$, *i.e. that state of each data variable is a value of the type of the variable,*
- *for each* $t \in T : s(t) = (assigned, accessibility, required)$ *where*
  - $assigned \in \{0, 1\}$ *determines whether a task is being executed,*
  - *accessibility is a function accesibility* $: Interface(t) \to \{visible, editable, hidden\}$ *determining for each data field of the task whether it is visible, editable or hidden, and*
  - *required is a function required* $: Interface(t) \to \{0, 1\}$ *determining for each data field of the task whether it is required (must have a value) or can be unfilled (without any value).*

Definition 3 (Events of a Petriflow OCP) *Events of a Petriflow OCP as defined in Definition 1 are defined as follows:*

- *for each* $t \in T$ *we define events:* $assign_t$, $cancel_t$ *and* $finish_t$ *and*

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VA/A VÝSKUMNÁ AGENTÚRA

- *for each $v \in V$ we define event get_v and a parametrized event set_v as a function with definition domain $Typing(v)$*

Definition 4 (Triggering events) *Given a Petriflow OCP as defined in Definition 1, and a set of users denoted $Users$ together with a mapping Role that associates each event as defined in Definition 3 with a subset of set $Users$, we define triggering of an event by a user $u \in Users$ in state s of the Petriflow OCP as follows:*

- *for each $t \in T$ event $assign_t$ can be triggered by user u in state $s \iff$*

An elementary net with (mixed) context is a five-tuple $N = (P, T, F, C_+, C_-)$, where (P, T, F) is an elementary net, and $C_+, C_- \subseteq P \times T$ are positive and negative context relations satisfying $(F \cup F^{-1}) \cap (C_+ \cup C_-) = C_+ \cap C_- = \emptyset$. For a transition t, ${}^\bullet t, t^\bullet$, ${}^+t$ and ${}^-t$ are defined as in the previous sections.

A transition t is enabled to occur in a marking m iff $({}^\bullet t \cup {}^+t) \subseteq m \wedge (m \setminus {}^\bullet t) \cap ({}^-t \cup t^\bullet) = \emptyset$. Its occurrence leads to the marking $m' = (m \setminus {}^\bullet t) \cup t^\bullet$, in symbols $m \xrightarrow{t} m'$.

Definition 5 (Place/transition nets with inhibitor arcs) *A p/t net with inhibitor arcs is a five-tuple $N = (P, T, F, W, C_-)$, where (P, T, F, W) is a p/t net, called the underlying p/t net of N, and $C_- \subseteq P \times T$ is an inhibitor relation (set of inhibitor arcs) satisfying $(F \cup F^{-1}) \cap C_- = \emptyset$. As usual, ${}^-t = \{p \mid (p, t) \in C_-\}$ for each $t \in T$. A marking of N is a multiset $m \in \mathbb{N}^P$.*

A transition t is enabled to occur in a marking m iff it is enabled to occur in m in the underlying p/t net and $\forall p \in P : p \in {}^-t \Rightarrow m(p) = 0$. The occurrence of an enabled transition t in a marking m is given as for the underlying p/t net.

Figure 4.1 shows an elementary net with (mixed) context.

Places and transitions of a net are also called elements of the net.

As usual, places are drawn as cycles, transitions as rectangles, and the flow relation is expressed using arcs connecting places and transitions.

Let (P, T, F) be a net and $x \in P \cup T$ be an element. The pre-set *${}^\bullet x$ is the set $\{y \in P \cup T \mid (y, x) \in F\}$, and the* post-set *$x^\bullet$ is the set $\{y \in P \cup T \mid (x, y) \in F\}$. Given a set $X \subseteq P \cup T$, this notation is extended as follows:*

$${}^\bullet X = \bigcup_{x \in X} {}^\bullet x \quad \textit{and} \quad X^\bullet = \bigcup_{x \in X} x^\bullet .$$

For technical reasons, we consider only nets in which every transition has a nonempty and finite pre-set and post-set.

Definition 6 (Place/Transition Net)

A place/transition net *(shortly* p/t net*) N is a quadruple (P, T, F, W), where (P, T, F) is a net and $W : F \to \mathbb{N}^+$ is a* weight function.

mixie-eps-converted-to.pdf

Figure 4.1: An elementary net with (mixed) context.

We extend the weight function W to pairs of net elements (x, y) satisfying $(x, y) \notin F$ by $W(x, y) = 0$.

To avoid confusion, sometimes we will write $N = (P_N, T_N, F_N, W_N)$ to denote $N = (P, T, F, W)$.

A marking *of a p/t net $N = (P, T, F, W)$ is a function $m : P \to \mathbb{N}$, i.e. a multiset over P. Graphically, a marking is expressed using a respective number of black tokens in each place.*

Definition 7 (Marked p/t-net)
A marked p/t-net *is a pair (N, m_0), where N is a p/t-net and m_0 is a marking of N called initial marking.*

4.3 Sequential semantics

Definition 8 (Occurrence rule)
Let $N = (P, T, F, W)$ be a p/t-net. A transition $t \in T$ is enabled to occur in a marking *m of N iff $m(p) \geq W(p, t)$ for each place $p \in {}^\bullet t$. If a transition t is enabled to occur in a marking m, then its* occurrence *leads to the new marking m' defined by $m'(p) = m(p) - W(p, t) + W(t, p)$ for each $p \in P$. We write $m \xrightarrow{t} m'$ to denote that t is enabled to occur in m and that its occurrence leads to m'.*

Definition 9 (Occurrence sequence, Reachability)
Let $N = (P, T, F, W)$ be a p/t-net and m be a marking of N. A finite sequence of transitions $\sigma = t_1 \dots t_n$, $n \in \mathbb{N}$ is called an occurrence sequence enabled in m and leading to m_n *iff there exists a sequence of markings $m_1, \dots, m_n$ such that*

$$m \xrightarrow{t_1} m_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} m_n.$$

The marking m_n is said to be reachable from the marking m.
In a marked p/t-net, markings reachable from the initial marking m_0 are shortly called reachable markings.

Chapter 5

Partial order based semantics

5.1 Step semantics

*In this section we recall the definition of step semantics for p/t nets. For more details see e.g. [**V92**].*

The occurrence of single transitions can be extended to the occurrence of multisets of transitions, called steps.

Definition 10 (Step occurrence rule)
Let $N = (P, T, F, W)$ be a p/t-net. A multiset of transitions $s \in \mathbb{N}^T$, called step, *is* enabled to occur in a marking m *of N iff*

$$\forall p \in P : m(p) \geq \sum_{t \in T} W(p,t) \cdot s(t).$$

If a step s is enabled to occur in a marking m, then its occurrence *leads to the new marking m' defined by*

$$\forall p \in P : m'(p) = m(p) + \sum_{t \in T} (W(t,p) - W(p,t)) \cdot s(t).$$

We write $m \xrightarrow{s} m'$ to denote that s is enabled to occur in m and that its occurrence leads to m'.

Definition 11 (Step sequence)
Let $N = (P, T, F, W)$ be a p/t-net and m be a marking of N. A finite sequence of steps (step sequence) $\sigma = s_1 \dots s_n$, $n \in \mathbb{N}$ is called a step sequence enabled in m and leading to m_n *if there exists a sequence of markings $m_1, \dots, m_n$ such that*

$$m \xrightarrow{s_1} m_1 \xrightarrow{s_2} \dots \xrightarrow{s_n} m_n.$$

We have the following proposition.

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VA/A VÝSKUMNÁ AGENTÚRA

Proposition 12 *The marking m' is reachable from the marking m if and only if there exists a step sequence enabled in m and leading to m'.*

5.2 Labelled partial orders

*In this section we recall the definition of semantics of p/t nets based on labelled partial orders, also known as partial words [**G81**] or pomsets [**P86**]. For the presented results see e.g. [**V92**].*

Definition 13 (Directed graph, (Labelled) partial order)
A directed graph *is a pair $(V, \to)$, where V is a finite* set of nodes *and $\to \subseteq V \times V$ is a binary relation over V called the* set of arcs*. As usual, given a binary relation $\to$ we write $a \to b$ to denote $(a, b) \in \to$.*

A partial order *is a directed graph $po = (V, <)$, where $<$ is an irreflexive and transitive binary relation on V.*

Two nodes v, v' of a partial order $(V, <)$ are called independent*, if $v \not< v'$ and $v' \not< v$. Denote $co \subseteq V \times V$ the set of all pairs of independent nodes of V. A* co-set *in a partial order $(V, <)$ is a subset $S \subseteq V$ fulfilling:*

$$\forall x, y \in S : \; x \, co \, y.$$

A slice *is a maximal co-set.*

If $v \, co \, v' \implies v = v'$, then we say that $<$ is a total order*. If the relation co is transitive, then we say that $<$ is* stepwise linearized.

For a co-set S of a partial order $(V, <)$ and a node $v \in V \setminus S$ we write:

- *$v < S$, if $v < s$ for an element $s \in S$, and*
- *$v \, co \, S$, if $v \, co \, s$ for all $s \in S$.*

If $<$ is stepwise linearized, we also write $S < S'$ for two slices S, S' of $<$ whenever $v < S'$ for an event $v \in S$.

Given partial orders $po_1 = (V, <_1)$ and $po_2 = (V, <_2)$, we say that po_2 is a sequentialization of *po_1 if $<_1 \subseteq <_2$. If a sequentialization po_2 of po_1 is a total order, we say that po_2 is a* linearization of *po_1 and if po_2 is stepwise linearized, we say that it is a* step linearization *of po_1.*

A labelled partial order *is a triple $lpo = (V, <, l)$, where $(V, <)$ is a partial order, and l is a* labelling function *on V. If X is a set of labels of lpo, i.e. $l : V \to X$, then for a slice $S \subseteq V$, we define the multiset $|S| \subseteq \mathbb{N}^X$ by*

$$\forall x \in X : \; |S|(x) = |\{v \in V \mid v \in S \wedge l(v) = x\}|.$$

We use the above notation defined for partial orders also for labelled partial orders.

Two labelled partial orders $(V_1, <_1, l_1), (V_2, <_2, l_2)$ *are isomorphic iff there exists a bijection* $\gamma : V_1 \to V_2$ *between nodes which preserves the partial order relation and the labelling function, i.e.* $\forall v_1, v_2 \in V : v_1 <_1 v_2 \iff \gamma(v_1) < \gamma(v_2) \wedge l(v_1) = l(\gamma(v_1))$.

Consider from now on a fixed p/t net $N = (P, T, F, W)$.

Obviously, the step sequences can be characterized by stepwise linearized labelled partial orders.

Definition 14 *Let* $\sigma = s_1 \dots s_n$ *(*$n \in \mathbb{N}$*) be a sequence of steps from* $\mathbb{N}^T$*. Then the stepwise linearized labelled partial order* $lpo_\sigma = (V, <, l)$ *with* $l : V \to T$ *and with slices* $S_1, \dots S_n$ *satisfying* $|S_i| = s_i$ *and* $i < j \Rightarrow S_i < S_j$ *for every* $i, j \in \{1, \dots n\}$ *is said to be associated to* σ.

*As it was observed in [**K88**], the step sequences can be used to define enabledness of labelled partial orders.*

Definition 15 *A labelled partial order* $lpo = (V, <, l)$ *with* $l : V \to T$ *is said to be* enabled to occur in a marking m *(shortly enabled in* m*) iff the following statement holds: Every step linearization of lpo is associated to a step sequence enabled to occur in* m.

Directly from the above definitions, we can also observe that the labelling of a linearization of an enabled labelled partial order is an occurrence sequence.

Remark 16 *The sequence of transitions* $\sigma = l(v_1) \dots l(v_n)$ *is an occurrence sequence enabled in* m *and leading to* m' *if and only if the total order* $(\{v_1, \dots, v_n\}, \prec, l)$ *satisfying* $\forall i, j \in \{1, \dots, n\} : i < j \Rightarrow v_i \prec v_j$ *is enabled in* m *and leads to* m'*. This total order is said to be associated to the occurrence sequence* σ.

Looking to the definition of enabledness of a step, we obtain the following proposition:

Proposition 17 *If a labelled partial order* $lpo = (V, <, l)$ *with* $l : V \to T$ *is enabled to occur in a marking* m *then the the following statement holds: For every co-set* C *of* $<$ *and every* $p \in P$*:*

$$m(p) + \sum_{v \in V \wedge v < C} (W(l(v), p) - W(p, l(v))) \geq \sum_{v \in C} W(p, l(v))$$

*Actually, the definition of enabledness can be reformulated considering only slices of labelled partial orders (for the proof see e.g. [**V92a**]).*

Proposition 18 *A labelled partial order* $lpo = (V, <, l)$ *with* $l : V \to T$ *is enabled to occur in a marking* m *if and only if the following statement holds: For every slice* S *of* $<$ *there exists a step linearization* lpo_S *of lpo associated to a step sequence enabled to occur in* m*, with* S *being a slice in* lpo_S.

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VAIA VÝSKUMNÁ AGENTÚRA

It is easy to observe that enabled labelled partial orders are closed w.r.t. sequentializations.

Proposition 19 *If a labelled partial order is enabled in m and leads to m', then every sequentialization of it is enabled in m and leads to m'.*

Special enabled labelled partial orders are those which are minimal w.r.t. inclusion.

Definition 20 (Enabled labelled partial order) *A labelled partial order $lpo = (V, <, l)$ enabled in m is said to be* minimal *iff there exists no labelled partial order $lpo' = (V, <', l)$ enabled in m with $<' \subset <$.*

We say that a set of labelled partial orders enabled to occur in m over the same set of events is compatible if the intersection of the labelled partial orders from this set is a labelled partial order enabled in m.

Definition 21 *Let m be a marking of N. Let $l : V \to T$ be a labelling and let $\lll$ be a set of partial orders on V satisfying: $(V, <, l)$ is a labelled partial order enabled to occur in m for every partial order $<$ from $\lll$.*

If the labelled partial order $lpo = (V, \prec = \cap_{<\in\lll}, l)$ is enabled to occur in m w.r.t. N, then we say that the set of labelled partial orders $\mathcal{C}_N^m\{(V, <, l) \mid <\in\lll\}$ is compatible w.r.t. N and m. *$\mathcal{C}_N^m$ and lpo are said to be* associated *with each other.*

The following proposition says that enabled labelled partial orders can be constructed by intersection of labelled partial orders associated to step sequences. In other words, for every enabled labelled partial order there exists an associated compatible set of labelled partial orders.

Proposition 22 *Let $lpo = (V, \prec, l)$ be a labelled partial order enabled to occur in m w.r.t. N. Then there exists a set X of labelled partial orders compatible w.r.t. N and m such that each labelled partial order from X is associated to a step sequence enabled to occur in m and X is associated to lpo.*

Directly from the enabledness of lpo, for every slice S of $(V, \prec)$ there exists a step sequence of N enabled to occur in m with associated labelled partial order $(V, <, l)$ enabled to occur in m satisfying: S is a slice of $(V, <)$ and $\prec \subseteq <$. Clearly, the intersection of these partial orders equals $\prec$, i.e. the set of these labelled partial orders is compatible w.r.t. N and m and associated to lpo.

In other words, the previous proposition together with the definition of enabledness says that every enabled labelled partial order can be constructed *from LPOs associated to step sequences.*

5.3 Processes and runs

Definition 23 (Occurrence net)
An occurrence net *is a net $O = (B, E, G)$ satisfying:*

(i) $|{}^\bullet b|, |b^\bullet| \leq 1$ *for every* $b \in B$ *(places are* unbranched*).*

(ii) O *is* acyclic*, i.e. the transitive closure* G^+ *of* G *is a partial order.*

Places of an occurrence net are called conditions *and transitions of an occurrence net are called* events.

The set of conditions of an occurrence net $O = (B, E, G)$ which are minimal (maximal) according to G^+ are denoted by $Min(O)$ ($Max(O)$). Clearly, $Min(O)$ and $Max(O)$ are slices w.r.t. G^+.

To avoid confusion, sometimes we will write $O = (B_O, E_O, G_O)$ to denote $O = (B, E, G)$.

Definition 24 (Process)
Let (N, m_0) be a marked p/t-net, with $N = (P, T, F, W)$. A process *of (N, m_0) is a pair $K = (O, \rho)$, where $O = (B, E, G)$ is an occurrence net and $\rho : B \cup E \to P \cup T$ is a labelling function, satisfying*

(i) $\rho(B) \subseteq P$ *and* $\rho(E) \subseteq T$.

(ii) $\forall e \in E, \forall p \in P : |\{b \in {}^\bullet e \mid \rho(b) = p\}| = W(p, \rho(e))$ *and*
$\forall e \in E, \forall p \in P : |\{b \in e^\bullet \mid \rho(b) = p\}| = W(\rho(e), p)$.

(iii) $\forall p \in P : |\{b \in Min(O) \mid \rho(b) = p\}| = m_0(p)$.

Two processes $K_1 = ((B_1, E_1, G_1), \rho_1)$ and $K_2 = ((B_2, E_2, G_2), \rho_2)$ are isomorphic (in symbols $K_1 \simeq K_2$) iff there exist bijections $\gamma : B_1 \to B_2, \delta : E_1 \to E_2$ such that $\forall b \in B_1, \forall e \in E_1 :$

$$(b, e) \in G_1 \iff (\gamma(b), \delta(e)) \in G_2,$$
$$(e, b) \in G_1 \iff (\delta(e), \gamma(b)) \in G_2,$$
$$\rho_1(b) = \rho_2(\gamma(b)), \rho_1(e) = \rho_2(\delta(e)).$$

Definition 25 (Run)
Let $K = (O, \rho)$ be a process of a marked p/t-net (N, m_0). The labelled partial order $lpo_K = (E, G^+|_{E \times E}, \rho|_E)$ is called run *of (N, m_0) representing K.*

A run $lpo = (E, <, \rho|_E)$ of (N, m_0) is said to be minimal *iff it there exists no other run $lpo' = (E, <', \rho|_E)$ of (N, m_0) with $<' \subset <$.*

*It is well known (see e.g. [***K88***; ***V92***; ***V92a***]) and easy to show from definition of processes that:*

Proposition 26 *Every run of (N, m_0) is enabled in m_0.*

From proposition 19 and proposition 26 follows:

Proposition 27 *If a labelled partial order is a sequentialization of a run of (N, m_0), then it is enabled in m_0.*

*The important result completing the relationship between enabled labelled partial orders and runs was proven in [***K88***; ***V92***; ***V92a***].*

Theorem 28 *If a labelled partial order is enabled in m_0 in a p/t net N, then it is a sequentialization of a run of the marked p/t net (N, m_0).*

As a consequence we obtain:

Theorem 29 *A run of (N, m_0) is minimal if and only if it is a minimal labelled partial order enabled in m_0.*

The previous theorem (together with the definition of enabledness of LPOs, compatible sets of enabled LPOs, and the fact that (minimal) enabled LPOs can be constructed from LPOs of step sequences) says that every minimal run can be constructed *from step sequences. In other words, LPOs associated to step sequences gives enough information about minimal runs, i.e. about the necessary causality between events in runs.*

5.4 From enabled LPOs to runs: Proof of Theorem 28

*As it was already mentioned, the fact that every enabled labelled partial orders includes a run was proven in [***K88***; ***V92***; ***V92a***]. As written in [***V92***], the proof in [***K88***] is quite complicated. On the other hand, the proof presented in [***V92***; ***V92a***] is based on a version of the marriage theorem from graph theory.*

We will give here our own (self contained) proof. In the proof we will use the so called flow property of a labelled partial order w.r.t. a marked place/transition net. The proof will split into two parts as shown in the following picture:

$$\begin{array}{rcl} \textit{1. Part: lpo is enabled in } m & \Longrightarrow & \textit{lpo fulfills the flow property} \\ \textit{2. Part: lpo fulfills the flow property} & \Longrightarrow & \textit{lpo is a sequentialization of a run.} \end{array}$$

5.4.1 Flow property

In order to simplify the formal definition of the flow property, let us define an extension of labelled partial orders $(V, <, l)$ *by adding an initial node which is smaller than all nodes from* V *and is labelled by a new label.*

Definition 30 (0-extension of a labelled partial order) *Let* $lpo = (V, <, l)$ *be a labelled partial order. Then a labelled partial order* $lpo^0 = (V^0, <^0, l^0)$*, where* $V^0 = (V \cup \{v_0\})$*,* $v_0 \notin V$*,* $<^0 = < \cup (\{v_0\} \times V)$*,* $l^0(v_0) \notin l(V)$ *and* $l^0|_V = l$*, is called* 0-extension of lpo.

Assigning non-negative natural numbers to the arcs of a 0*-extension of a labelled partial order we define a so called flow (function) of this labelled partial order (with* v_0 *as its source).*

Definition 31 (Flow function of a labelled partial order) *Let* $lpo = (V, <, l)$ *be a labelled partial order and* $lpo^0 = (V^0, <^0, l^0)$ *be a* 0*-extension of* lpo*. A function* $x : <^0 \to \mathbb{N}_0$ *is called* flow function*, or simply* flow*, of* lpo*. For* $v \in V$*, we denote*

- $\sum_{v' <^0 v} x((v', v))$ *the* ingoing flow of v w.r.t. x*, and*
- $\sum_{v <^0 v'} x((v, v'))$ *the* outgoing flow of v w.r.t. x.

If the labelled partial order $lpo = (V, <, l)$ *is a run representing a process of a marked p/t-net, then for every place* p *there is a flow, called* canonical flow of the run w.r.t. p*, which counts for every* $v < v'$ *the number of post-conditions of* v *labelled by* p *which are pre-conditions of* v' *in the process. The outgoing flow of the source event* v_0 *represents the the number of minimal conditions labelled by* p *which are used by further events. By definition, this canonical flow of a run w.r.t. a place* p *respects the weight function and the initial marking of the underlying marked p/t-net in the following sense:*

(A) The ingoing flow of an event v *equals the number of tokens consumed from place* p *by the occurrence of transition* $l(v)$*.*

(B) The outgoing flow of an event v *is less or equal to the number of tokens which are produced by the occurrence of transition* $l(v)$ *in place* p*. In particular, the outgoing flow of the source event* v_0 *is less or equal to the number of tokens in place* p *of the initial marking* m_0*.*

Thus, the canonical flow of a run lpo abstracts from the individuality of conditions and encodes the flow relation G *of the process by natural numbers.*

In general, we say that an arbitrary labelled partial order, whose labels are transitions of a marked p/t-net, fulfils the flow property w.r.t. this marked p/t-net, if for every place there exists a flow which fulfills the properties (A) *and* (B)*.*

Definition 32 (Flow property) *Let $lpo = (V, <, l)$ be a labelled partial order with $l(V) = T$ and let $lpo^0 = (V^0, <^0, l^0)$ be a 0-extension of lpo. Denote $post(l(v_0)) = m_0$. We say that lpo fulfils the flow property w.r.t. (N, m_0) if the following statement holds: For every place $p \in P$ there exists a flow $x_p : <^0 \to \mathbb{N}_0$ such that*

(A) For every $v' \in V$:

$$\sum_{v <^0 v'} x_p(v, v') = pre(l(v'))(p).$$

(B) For every $v \in V^0$:

$$\sum_{v <^0 v'} x_p(v, v') \leq post(l(v))(p).$$

As we will show in the second part of the proof, the flow property is a sufficient condition for a labelled partial order to be a sequentialization of a run.

Thus, the crucial part of the proof is the first part, which shows that the enabledness implies the flow property.

Because the flow property of a labelled partial order is (for simplicity reasons) defined using its 0-extension, we start with the following remark:

Remark 33 *Observe that given a labelled partial order $(V, \prec, l)$ and its 0-extension $(V^0, \prec^0, l^0)$, $\prec^0$ preserves the co-sets of $\prec$. More exactly, every co-set of $\prec$ is a co-set of $\prec^0$ and the only co-set of $\prec^0$, which is not a co-set of $\prec$, is the one-element set $\{v_0\}$.*

5.4.2 First part: From enabledness to the flow property

In this part of the proof, we have to prove the following statement.

Theorem 34 *If a labelled partial order $lpo = (V, \prec, l)$ is enabled in m_0 w.r.t. (N, m_0), then it fulfils the flow property w.r.t. (N, m_0).*

This implication is the central step in the proof of Theorem 28. Notice that it is a special optimization problem.

Given $v \in V$, the sum $\sum_{v <^0 v'} x(v, v')$ in the left side of equality (A) in the definition of the flow property will be referred to as the (A)-sum of x for v *and similarly, given $v \in V^0$, the sum $\sum_{v' <^0 v} x(v', v)$ in the left side of inequality (B) will be referred to as the (B)*-sum of x for v.

We will show the theorem by contradiction, i.e. we assume that there is a place p_0 for which there does not exist a flow $x_{p_0} : <^0 \to \mathbb{N}_0$ such that

(A) For every $v' \in V$:

$$\sum_{v <^0 v'} x_{p_0}(v, v') = pre(l(v'))(p_0).$$

(B) For every $v \in V^0$:

$$\sum_{v <^0 v'} x_{p_0}(v, v') \leq post(l(v))(p_0).$$

Clearly, for every $p \in P$ there exists a flow function fulfilling (A) (e.g. the function $x_p(v_0, v) = pre(l(v))(p)$ for every $v \in V$ and $x_p(v, v') = 0$ for every $v \prec v'$, $v \neq v_0$).

In the following, denote $V = \{v_1, \ldots, v_{|V|}\}$ such that $v_i \prec v_j$ implies $i < j$.

Consider the set X of all flow functions x fulfilling (A) for the place p_0. By assumption, none of these functions fulfil (B). We say that a function $x \in X$ does not fulfil (B) for an index i, *if $\sum_{v_i \prec^0 v_j} x(v_i, v_j) > post(l(v_i))(p_0)$.*

Denote k_x the smallest index for which a flow function $x \in X$ does not fulfil (B). Define the set $X_{max} \subset X$ as the non-empty set of all flow functions $x \in X$ which maximize k_x:

(i) For all functions $x' \in X$: $k_{x'} \leqslant k_x$.

Let $max = k_x$ be the smallest index for which a function $x \in X_{max}$ does not fulfil (B).

Now, choose a function $x \in X_{max}$ which minimizes the (B)-sum of x for v_{max}:

(ii) For all functions $x' \in X_{max}$

$$\sum_{v_{max} \prec^0 v_m} x(v_{max}, v_m) \leq \sum_{v_{max} \prec^0 v_m} x'(v_{max}, v_m).$$

Let $MIN_{max} = \sum_{v_{max} \prec^0 v_m} x(v_{max}, v_m)$ be the (B)-sum of x.

Thus, the assumption that the flow property does not hold for a place p_0 implies that there exists a function x which fulfils (i), (ii) and (A), i.e. a function x for which

$$\begin{aligned} MIN_{max} = \sum_{v_{max} \prec^0 v_j} x(v_{max}, v_j) \quad &> \quad post(l(v_{max}))(p_0), \\ \forall i < max : \sum_{v_i \prec^0 v_j} x(v_i, v_j) \quad &\leqslant \quad post(l(v_i))(p_0), \\ \forall j : \sum_{v_i \prec^0 v_j} x(v_i, v_j) \quad &= \quad pre(l(v_j))(p_0). \end{aligned}$$

Proposition 17 gives a necessary condition for enabledness. To get a contradiction with the enabledness of lpo, our aim is to show that, if such a function x exists, then we can construct a co-set C of $\prec$ for which the necessary condition formulated in Proposition 17 does not hold, i.e.

$$(*) \quad m_0(p_0) + \sum_{v \in V \wedge v \prec C} (post(l(v))(p_0) - pre(l(v))(p_0)) - \sum_{v \in C} pre(l(v))(p_0) < 0.$$

From now on, fix a function x fulfilling the properties (i) and (ii). We will construct C in such a way that

(I) $v_{max} \prec C$ *and* $v_i \prec^0 C \Rightarrow i \leqslant max$,

(II) each value in the left side of $(*)$ *equals a sum over x-flows, except* $post(l(v_{max}))(p_0) < \sum_{v_{max} \prec^0 v_j} x(v_{max}, v_j)$, *and*

(III) when replacing these values by the respective x-flow sums, all flows which are not 0 *are counted once negatively and once positively. That means, whenever* $x(v, v') > 0$ *and* $v \prec^0 C$ *then* $v' \prec^0 C$ *or* $v' \in C$.

We will need an important property of the function x. The simple version of this property says the following: If an event v_l consumes something from v_{max} then v_l is not greater than any event v_j with index $j > max$.

Lemma 35 *For all* $v_j, v_l \in V$*: If* $x(v_{max}, v_l) > 0$ *and* $max < j$ *then* $v_j \not\prec^0 v_l$.

Assume $v_j \prec^0 v_l$. *We construct a function* $x' \in X$ *contradicting that x fulfills the conditions* (i) *and* (ii), *by modifying x as follows:*

$$\begin{aligned} x'(v_j, v_l) &= x(v_j, v_l) + 1, \\ x'(v_{max}, v_l) &= x(v_{max}, v_l) - 1. \end{aligned}$$

First observe that by construction x' fulfills property (A), since the (A)-sums of x and x' are equal, i.e. $x' \in X$.

We can distinguish two cases: Either the smallest indexes for which x' and x do not fulfil property (B) are equal, i.e. $k_{x'} = k_x = max$. In this case, both x and x' are in X_{max}, but the (B)-sum of x' for v_{max} is less than the (B)-sum of x for v_{max}. Thus, x' would decrease MIN_{max}. This contradicts that x fulfills (ii).

The other possibility is $k_{x'} \neq max$. This may be the case, since the (B)-sum of x' for v_{max} is less than the (B)-sum of x for v_{max}, resp. since the (B)-sum of x' for v_j greater than the (B)-sum of x for v_j. From $j > max$ follows $k_{x'} > max$, i.e. x' would increase max. This contradicts that x fulfills (i).

Remark 36 *Observe that Lemma 35 is equivalent to the statement that each event, which is smaller than an event taking something from v_{max}, has an index smaller than max:*

$$\forall v_l, v_j \in V : \; (x(v_{max}, v_l) > 0) \wedge (v_j \prec^0 v_l) \Rightarrow (j \leqslant max).$$

Moreover, Lemma 35 implies that events taking something from v_{max} are independent:

$$\forall v_l, v_m \in V : \; (x(v_{max}, v_l) > 0) \wedge (x(v_{max}, v_m) > 0) \Rightarrow (v_m \, co \, v_l).$$

Denote by $V_{max} = \{v_l \mid x(v_{max}, v_l) > 0\}$ the set of events which take something from v_{max}. By the previous remark, V_{max} is a co-set of $\prec$.

Moreover, the (B)-sum of x for v equals the value $post(l(v))(p_0)$ for every event $v \in V^0$ which is smaller than V_{max}:

Lemma 37 *For all $v_j \in V^0$: If $v_j \neq v_{max}$ and $v_j \prec^0 V_{max}$, then the (B)-sum of v_j equals $post(l(v_j))(p_0)$:*

$$\sum_{v_j \prec^0 v_i} x(v_j, v_i) = post(l(v_j))(p_0).$$

By Lemma 35, $v_j \prec^0 V_{max}$ implies $j < max$. Assume $\sum_{v_j \prec^0 v_i} x(v_j, v_i) \neq post(l(v_j))(p_0)$. By the definition of max, we get $\sum_{v_j \prec^0 v_i} x(v_j, v_i) < post(l(v_j))(p_0)$. We construct a function $x' \in X$ contradicting that x fulfills the conditions (i) and (ii), by modifying x as follows: Choose $v_l \in V_{max}$ with $v_j \prec^0 v_l$ and set

$$\begin{aligned} x'(v_j, v_l) &= x(v_j, v_l) + 1, \\ x'(v_k, v_l) &= x(v_k, v_l) - 1. \end{aligned}$$

As in the proof of Lemma 35, x' would either decrease MIN_{max} or increase max (since although the (B)-sum of x' for v_j equals that one of x increased by 1, this would not decrease $k_{x'}$ compared to $k_x = max$ due to the assumption of the proof).

This is a motivation to choose $C = V_{max}$ as a first approximation, since this gives (I) and (II). But this is not enough, i.e. V_{max} must be extended: Taking $v_i \prec^0 V_{max}$, there may be a flow $x(v_i, v_m) > 0$ from v_i, with $v_m \notin V_{max}$. That means (III) does not hold yet. In order to get (III), we have to add such events v_m to V_{max}. First we need to see that V_{max} extended in such a way is still a co-set. This follows from an extended version of the property given by Lemma 35, which says the following: If *an event v_l takes something from v_{max} and is greater than an event v_i from which an event v_m takes something,* then *v_m is not greater than any event v_j with index $j \geqslant max$.*

Lemma 38 *For all $v_i, v_j, v_l, v_m \in V^0$: if $x(v_{max}, v_l) > 0$, $v_i \prec^0 v_l$, $x(v_i, v_m) > 0$ and $max < j$, then $v_j \not\prec^0 v_m$.*

From Lemma 35 follows $i \leq max$. Assume $v_j \prec^0 v_m$. Observe that we have $max < j < m$, $max < l$ and $i < l$. In the case that $l = m$ the property becomes the same as in Lemma 35. In the case that $i = max$ the property is also fulfilled, just considering v_m in the role of v_l in Lemma 35. It may be the case that $j = l$, but this does not influence the proof.

In case $i \neq max$ and $l \neq m$ we construct (as in Lemma 35) a function $x' \in X$ contradicting that x fulfills the conditions (i) and (ii), by modifying x:

$$\begin{aligned} x'(v_j, v_m) &= x(v_j, v_m) + 1, \\ x'(v_i, v_m) &= x(v_i, v_m) - 1, \\ x'(v_i, v_l) &= x(v_i, v_l) + 1, \\ x'(v_k, v_l) &= x(v_k, v_l) - 1. \end{aligned}$$

As in the proof of Lemma 35 one may verify, that $x' \in X$ and either MIN_{max} would be decreased or max would be increased.

We will first consider a special case, where the set of events smaller than C is not extended by adding v_m. Then we will once more extend the properties given in Lemmas 35 and 38 and construct such C for the general case.

A special case

Consider first the special case, where every event with index smaller than max is also smaller than an event from V_{max}, i.e.

$$(\blacktriangleleft) \quad \forall i \in \{0, \ldots, max\} : v_i \prec^0 V_{max}.$$

In this case, we construct C as explained above: namely, add to V_{max} all events not smaller then V_{max}, which take something from events smaller then V_{max}:

$$C = V_{max} \cup \{v_m \mid v_m \not\prec^0 V_{max} \wedge \exists v_i \prec^0 V_{max} : x(v_i, v_m) > 0\}.$$

Using Lemma 38 we will show that C is a co-set and fulfils the property

$$(**) \quad \{v \mid v \prec^0 C\} = \{v \mid v \prec^0 V_{max}\}.$$

Together with Lemma 35 this gives that C preserves the properties $(I), (II)$ and (III), and therefore $(*)$ holds.

Observe, that every event v_j from C has an index greater than max (according to $(\blacktriangleleft)$). Per definition $v_i \prec^0 V_{max}$ means that there exists $v_l \in V_{max}$, (i.e. $x(v_{max}, v_l) > 0$), such that $v_i \prec^0 v_l$. By Lemma 38 every event v_m added to C is not greater than any $v_j \in C$. This implies, together with the fact that V_{max} itself is a co-set, that any pair of events from C is independent and therefore C is a co-set of $\prec^0$, too. According to Remark 33, C is a co-set of $\prec$.

Moreover, Lemma 38 says generally that no event in the co-set C is greater than any event with index greater than max, i.e. events from C are greater only as events which are smaller than events in V_{max}. Thus, C is a co-set fulfilling $(**)$.

Now, observe that by Lemma 37

- *the value* $m_0(p_0)$ *can be replaced by* $\sum_{v_0 \prec^0 v'} x(v_0, v')(p_0)$ *in* $(*)$*, and*
- *the value* $post(l(v))(p_0)$ *can be replaced by* $\sum_{v \prec^0 v'} x(v, v')(p_0)$ *in* $(*)$ *for* $v \neq v_k$ *.*

Moreover, the values $pre(l(v))(p_0)$ *can be replaced by* $\sum_{v' \prec^0 v} x(v', v)$ *due to condition* (A)*. We can also use the fact that there is no event smaller than* v_0 *and therefore* $\sum_{v' \prec^0 v_0} x(v', v) = 0$*.*

Since x *does not satisfy* (B) *for the index max we have also*

$$\sum_{v_{max} \prec^0 v'} x(v_{max}, v') > post(l(v_{max}))(p_0).$$

Altogether, we get:

$$m_0(p) + \sum_{v \in V \wedge v \prec C} (post(l(v))(p) - pre(l(v))(p)) - \sum_{v \in C} pre(l(v))(p) <$$
$$\sum_{v \in V^0 \wedge v \prec^0 C} \left(\sum_{v \prec^0 v'} x(v, v') - \sum_{v' \prec^0 v} x(v', v) \right) - \sum_{v \in C \wedge v' \prec^0 v} x(v', v).$$

Observe that the right side of the inequality equals 0*, because in the sum each value* $x(v, v')$ *either equals* 0 *or is counted once positively and once negatively, i.e.*

- $v \prec^0 C \wedge v \prec^0 v' \wedge x(v, v') > 0 \Rightarrow v' \prec^0 C \vee v' \in C$ *(for positive values).*
- $v \prec^0 C \wedge v' \prec^0 v, \wedge x(v', v) > 0 \Rightarrow v' \prec^0 C$ *(for negative values).*
- $v \in C \wedge v' \prec^0 v \wedge x(v', v) > 0 \Rightarrow v' \prec^0 C$ *(for negative values).*

Thus, for the case $(\blacktriangleleft)$ *is satisfyed.*

The general case

Now, we will construct C *satisfying* $(*)$ *for the general case, when* $(\blacktriangleleft)$ *does not necessarily hold. In this case, denote*

$$Q = \{v_i \in V^0 \mid i < max \wedge v_i \not\prec^0 V_{max}\}$$

the set of events which are not smaller than V_{max}*, but have smaller index than* max*.*

First, we will construct the set of events W *which are smaller than the searched co-set* C*. We start with the set of events* U *smaller than* V_{max}*:*

$$U = \{v \in V^0 \mid v \prec^0 V_{max}\}.$$

Now, we recursively add to this set events from V^0 *in such a way that each event, which is smaller than an event taking something from (an event in) the resulting set* W*, also belongs to* W*:*

- $\forall v \in V^0 : v \in U \Rightarrow v \in W$
- $\forall v_j, v_m \in V : (v_j \prec^0 v_m \wedge \exists v_i \in W : x(v_i, v_m) > 0) \Longrightarrow (v_j \in W)$

Observe that, because $\prec^0$ *is transitive, the set* W *is closed under* $\prec^0$*, i.e.* $\forall v, v' \in V^0 : (v \prec^0 v' \wedge v' \in W) \Rightarrow (v \in W)$*.*

Now, we define C *as the set of all events, which take something from (an event in)* W*:*

$$C = \{v \mid v \not\prec^0 V_{max} \wedge \exists v' \in W : x(v', v) > 0\} \; (\supseteq V_{max}).$$

From the extended versions of Lemma 38 and Lemma 37, we get that C *satisfies* (I)*-*(III)*. First we show statement* (I)*:*

Lemma 39 *Each* $v_j \in W$ *satisfies* $j \leq max$*, i.e.* $W \subseteq U \cup Q$*.*

Suppose $v_j \in W \setminus (U \cup Q)$*, i.e.* $j > max$*.*

According to the definitions of U *and* W *there exists* v_m *with* $v_j \prec^0 v_m$ *and a sequence of events* $v^0 = v_{max}, v^1, v^{1'}, v^2, v^{2'}, \ldots, v^k = v_j, v^{k'} = v_m$ *such that:*

- $v^i \neq v^j$ *and* $v^{i'} \neq v^{j'}$ *for* $i \neq j$*,*
- $v^i \in W$ *for* $0 \leq i \leq k$*,*
- $x(v^i, v^{(i+1)'}) > 0$ *for* $0 \leq i < k$ *and*
- $v^i \prec^0 v^{i'}$ *for* $1 \leq i \leq k$*.*

Thus, we can construct a function x' *from* x*, which contradicts that* x *satisfies* (i) *and* (ii) *(similarly as in the proof of Lemma 38):*

- $x'(v^i, v^{(i+1)'}) = x(v^i, v^{(i+1)'}) - 1$ *for* $0 \leq i < k$*,*
- $x'(v^i, v^{i'}) = x(v^i, v^{i'}) + 1$ *for* $1 \leq i \leq k$*.*

With the same argumentation as in the proof of the previous lemma, we get an extended version of Lemma 38. This gives that C *is a co-set of* $\prec$ *and implies* (III)*.*

Lemma 40 *For all* $v_i, v_j, v_m \in V^0$*: if* $v_i \in W$*,* $x(v_i, v_m) > 0$ *and* $max < j$*, then* $v_j \not\prec^0 v_m$*.*

Finally, also Lemma 37 can be generalized in the same way, which gives (II)*.*

Lemma 41 *For all* $v_j \in W$*: if* $v_j \neq v_{max}$*, then the* (B)*-sum of* v_j *equals* $post(l(v_j))(p_0)$*:*

$$\sum_{v_j \prec^0 v_i} x(v_j, v_i) = post(l(v_j))(p_0).$$

By a similar argumentation as in the special case, we deduce from Lemma 40 and Lemma 41 that C satisfies

$$(*)\quad m_0(p_0)+\sum_{v\prec C}(post(l(v))(p_0)-pre(l(v))(p_0))-\sum_{v\in C}pre(l(v))(p_0)<0.$$

This concludes the proof of Theorem 34 for the general case.

5.4.3 Second Part: From flow property to run inclusion

In this part we will prove that for every labelled partial order $(V,\prec,l)$ which fulfils the flow property w.r.t. (N,m_0) we can construct a process represented by a run $(V,<,l)$ of (N,m_0) such that $<\subseteq\prec$.

Lemma 42 *Let $(V,\prec,l)$ be a labelled partial which fulfils the flow property w.r.t. (N,m_0). Then there exists a run $(V,<,l)$ of (N,m_0) such that $<\subseteq\prec$.*

From the definition of the flow property, for every $p\in P$ there exists a function x_p which fulfils (A) and (B). We will fix these functions and use them to construct a process $K=(O,\rho)$ of (N,m_0) with $O=(B,V,G)$ and $\rho|_V=l$, satisfying $<=G^+|_{V\times V}\subseteq\prec$. According to the definition of runs, this will conclude the proof.

For convenience, denote $V=\{v_1,\dots,v_{|V|}\}$ such that $v_i\prec v_j$ implies $i<j$.

First define the set of conditions and the labelling of conditions.

For every event $v\in V^0$ we define the set of post-conditions of v labelled by $p\in P$:

$$B_p^v=\{p_v^1,\dots p_v^{post(l(v))(p)}\}.$$

Thus, the number of these post-conditions equals the value $post(l(v))(p)$. Especially, the number of post-conditions of v_0 labelled by $p\in P$ equals $m_0(p)$.

Denote $B_p=\cup_{v\in V^0}B_p^v$ the set of all conditions labelled by p.

Define the labelling of conditions by $\rho(b)=p$ for $b\in B_p$.

Finally, the set of all conditions of the process is given by $B=\cup_{p\in P}B_p$.

It remains to define the flow relation G. It is the union of all ingoing and outgoing arcs of all events $v\in V$.

An event $v\in V$ has an outgoing arc to each of its post-conditions (observe that $v_0\notin V$). Thus, the set of outgoing arcs of an event $v\in V$ labelled by $p\in P$ is

$$G_p^{v^\bullet}=v\times B_p^v.$$

The ingoing arcs are defined w.r.t. the flows. If $x_p(v,v_m)>0$, then we connect exactly $x_p(v,v_m)$ post-conditions of v labelled by p with v_m. In order to avoid branching of conditions, we connect post-conditions $p_v^1,\dots,p_v^{x_p(v,v_m)}$ with v_m which has the smallest *index m from all events v_m with $x_p(v, v_m) > 0$, and so on. Formally, define the set of ingoing arcs from conditions labelled by $p \in P$ to an event $v_m \in V$ by*

$$\begin{aligned} G_p^{\bullet v_m} &= \{(p_v^i, v_m) \mid v \in V_0,\, x_p(v, v_m) > 0, \\ &\sum_{j<m} x_p(v, v_j) < i \leq \sum_{j \leq m} x_p(v, v_j)\}. \end{aligned}$$

Because x_p fulfils (B), i.e. the number of post-conditions of an event $v \in V^0$ is not less than the outgoing flow of v, by this construction any event $v_m \in V$ is connected with exactly $x_p(v, v_m)$ post-conditions of v labelled by p whenever $x_p(v, v_m) > 0$. Because of this and because x_p also fulfils (A), by this construction every $v_m \in V$ has exactly $pre(l(v_m))(p)$ pre-conditions labelled by $p \in P$.

Finally denote $G_p = \cup_{v \in V^0}(G_p^{\bullet v} \cup G_p^{v^\bullet})$ for every $p \in P$ and $G = \cup_{p \in P} G_p$.

By construction, the conditions are unbranched and the defined net is acyclic, i.e. $O = (B, V, G)$ is an occurrence net. By the previous construction and argumentation, $K = (O, \rho)$ is a process of (N, m_0).

It remains to show that $<= G^+|_{V \times V} \subseteq \prec$. Denote

$$R = \{(v, v') \in V \times V \mid v^\bullet \cap {}^\bullet v' \neq \emptyset\}.$$

Observe that $G^+|_{V \times V} = R^+$. Observe that by construction of G we have

$$(v, v') \in R \Longrightarrow (\exists p \in P : x_p(v, v') > 0).$$

Because $x_p(v, v') > 0$ implies $v \prec v'$ and $\prec$ is transitive, this gives $<= G^+|_{V \times V} \subseteq \prec$.

5.5 Summary

In order to summarize the relationships between occurrence sequences, step sequences, enabled LPOs and processes stated above we can compare the related sets of LPOs associated to them.

Given a p/t net N and a marking m_0, let us denote:

- *the set of (isomorphism classes of) LPOs associated to occurrence sequences enabled in m_0 by* **SEQ**,
- *the set of (isomorphism classes of) LPOs associated to step sequences enabled in m_0 by* **STEPSEQ**,
- *the set of (isomorphism classes of) LPOs enabled to occur in m_0 by* **ENABLED** *and the set of (isomorphism classes of) minimal LPOs enabled to occur in m_0 by* **MINENABLED**

- *the set of (isomorphism classes of) runs of* (N, m_0) *by* **RUN** *and the set of (isomorphism clsses of) minimal runs of* (N, m_0) *by* **MINRUN**.

.

The relationship between these sets w.r.t. set inclusion is given as follows:

$$\mathbf{SEQ} \subseteq \mathbf{STEPSEQ} \subseteq \mathbf{ENABLED}$$

$$\mathbf{RUN} \subseteq \mathbf{ENABLED}$$

Another important relationship between these sets is the relationship w.r.t. sequentialization. Taking two sets X, Y *of LPOs, we denote by* $X \Supset Y$ *the fact that each LPO from* X *is a sequentialization of an LPO from* Y*, i.e. each LPO from* X *includes an LPO from* Y*:*

$$\mathbf{SEQ} \Supset \mathbf{STEPSEQ} \Supset \mathbf{ENABLED} \Supset \mathbf{RUN}$$

As a consequence:

$$\mathbf{MINENABLED} = \mathbf{MINRUN}$$

Importantly, enabled labelled partial orders and therefore also minimal runs can be constructed from LPOs associated to step sequences.

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VAIA VÝSKUMNÁ AGENTÚRA

Chapter 6

Commutative processes and runs

*As mentioned in the introduction to this part, one occurrence sequence can be in general a sequentialization of two different processes. On the other hand, there are in general many occurrence sequences, which are sequentializations of one process. One may wonder if there exists an equivalence on occurrence sequences and an equivalence on processes, which will respect the relation "being a sequentialization" between occurrence sequences and processes in the following sense: two occurrence sequences are equivalent if and only if they are sequentializations of equivalent processes. This question is investigated in [**BD87**]: For finite occurrence sequences and processes with a finite number of events, which are of interests in this work, such equivalences are identified and shown to be the finest equivalences with the property.*

Definition 43 (Swapping) *Let (N, m_0) be a marked p/t-net, with $N = (P, T, F, W)$. Let $K = (O, \rho)$, be a process of (N, m_0) with $O = (B, E, G)$. Let $b_1, b_2 \in B$, $b_1 \, co \, b_2$ and $\rho(b_1) = \rho(b_2)$. We define $G_1 = \{(b_1, e) \mid (b_2, e) \in G\}$ and $G_2 = \{(b_2, e) \mid (b_1, e) \in G\}$. Then we define $G' = G_1 \cup G_2 \cup (G \cap (E \times B)) \cup (G \cap ((B \setminus \{b_1, b_2\}) \times E))$. Thus, G' is obtained from G by interchanging arcs from b_1 and b_2. Finally, we define $swap(K, b_1, b_2) = ((B, E, G'), \rho)$.*

*As it is shown in [**BD87**]:*

Theorem 44 *Let $K = (O, \rho)$ be a process of a marked p/t net (N, m_0) with $O = (B, E, G)$. Let $b_1, b_2 \in B$, $b_1 \, co \, b_2$ and $\rho(b_1) = \rho(b_2)$. Then $swap(K, b_1, b_2)$ is a process of (N, m_0).*

Definition 45 *Let $K_1 = ((B, E, G), \rho)$ and K_2 be processes of a marked p/t net (N, m_0). Then we define $K_1 \equiv_1 K_2$ if there are conditions $b_1, b_2 \in B$ such that $b_1 \, co \, b_2$ and $\rho(b_1) = \rho(b_2)$ and K_2 is (isomorphic to) $swap(K_1, b_1, b_2)$.*

It is easy to see that $\equiv_1$ is symmetric. Thus, $\equiv_1^*$ is an equivalence relation on processes of (N, m_0).

Definition 46 *The equivalence relation $\equiv_1^*$ on processes of (N, m_0) is called* swapping equivalence*. The equivalence classes of processes w.r.t. the swapping equivalence are called* commutative processes *of (N, m_0).*

*The searched equivalence on occurrence sequences is defined in [**BD87**] using a relation $\equiv_0$ as follows:*

Definition 47 (Exchange relation $\equiv_0$ on occurrence sequences) *Let N be a p/t net and m_0 be a marking of N. Let $\sigma_1 = t_1 \ldots t_{i-1} t_i t_{i+1} t_{i+2} \ldots t_n, \sigma_2 = t_1 \ldots t_{i-1} t_{i+1} t_i t_{i+2} \ldots t_n$ be occurrence sequences of N enabled to occur in m_0. Then $\sigma_1 \equiv_0 \sigma_2$ iff $\sigma = \{t_1\} \ldots \{t_{i-1}\}\{t_i, t_{i+1}\}\{t_{i+2}\} \ldots$ is a step sequence of N enabled to occur in m_0.*

Again, it is easy to see that $\equiv_0$ is symmetric and therefore $\equiv_0^*$ is an equivalence relation.

Definition 48 *The equivalence relation $\equiv_0^*$ on occurrence sequences of N enabled to occur in m_0 is called* exchange equivalence.

*The relationship between exchange equivalence classes and swapping equivalence classes proven in [**BD87**] says:*

Theorem 49 *Let N be a p/t net and m_0 a marking of N. Let σ_1, σ_2 be occurrence sequences of N enabled to occur in m_0 and let K_1, K_2 be processes of (N, m_0) such that the total labelled partial order associated to σ_i is a linearization of the run representing K_i ($i \in \{1, 2\}$). Then $\sigma_1 \equiv_0^* \sigma_2$ if and only if $K_1 \equiv_1^* K_2$.*

*Moreover, as it is proved in [**BD87**] for the finite case, $\equiv_0^*$ and $\equiv_1^*$ are the finest equivalences which satisfy the above result: These equivalences partition the set of occurrence sequences and processes respectively into finest equivalence classes such that the relation "being a sequentialization" define a bijection on these classes.*

As a consequence, these result extend on runs and enabled labelled partial orders. The relation $\frown$ on runs, relating runs if and only if the processes represented by these runs are swapping equivalent, is an equivalence relation. Similarly, the relation $\smile$ on the set of all labelled partial orders enabled in a fixed marking, which relates these labelled partial orders if and only if some of their linearizations are associated to exchange equivalent occurrence sequences, is an equivalence relation.

Definition 50 *Let K_1, K_2 be processes of a marked p/t net (N, m_0) and let lpo_1, lpo_2 be runs representing K_1, K_2, respectively. Then the equivalence relation $\frown$ on runs given by $lpo_1 \frown lpo_1 \iff K_1 \equiv_1^* K_2$ is called swapping equivalence on runs of (N, m_0). The equivalence classes of runs w.r.t. the swapping equivalence are called* commutative runs *of (N, m_0).*

Definition 51 *Let N be a p/t net and m_0 be a marking of N. Let lpo_1, lpo_2 be labelled partial orders enabled to occur in m_0. Then the equivalence relation $\smile$ on the set of all labelled partial orders enabled to occur in m_0 given by:*

- *$lpo_1 \smile lpo_1$ if and only if there exists occurrence sequences σ_1, σ_2 such that the total labelled partial order associated to σ_i is a linearization of lpo_i ($i \in \{1, 2\}$) and $\sigma_1 \equiv_0^* \sigma_2$*

is called exchange equivalence on labelled partial orders of N enabled to occur in m_0. The equivalence classes of labelled partial orders enabled to occur in m_0 w.r.t. the exchange equivalence are called commutative labelled partial orders *enabled to occur in m_0.*

From Theorem 49 and the results on relationships between runs and enabled labelled partial orders we get:

Theorem 52 *Let N be a p/t net and m_0 a marking of N. Let lpo_1, lpo_2 be labelled partial orders enabled to occur in m_0 and let lpo'_1, lpo'_2 be runs of (N, m_0) such that lpo_i is a sequentialization of the run lpo'_i ($i \in \{1, 2\}$). Then $lpo_1 \smile lpo_2$ if and only if $lpo'_1 \frown lpo'_2$.*

Thus, the equivalences $\smile, \frown$ partition the set of enabled labelled partial orders and runs respectively into commutative enabled labelled partial orders and commutative runs such that the relation "being a sequentialization" define a bijection between them.

Chapter 7

Semantics of algebraic p/t nets

7.1 Algebraic p/t nets

Definition 53 (Algebraic p/t-net)
An algebraic p/t-net *is a quadruple* $N = (P, T, pre, post)$*, where* P *is a finite* set of places*,* T *is a finite* set of transitions*, satisfying* $P \cap T = \emptyset$*, and* $pre, post : T \to \mathbb{N}^P$ *are* source *and* target functions*, respectively.*

*The formal definition of algebraic p/t-nets was introduced in [***MM90***]. We write* $t : m \to m'$ *to denote that* $t \in T$*,* $pre(t) = m$ *and* $post(t) = m'$*.*

Definition 54 (Process term semantics, [MM90])
Given an algebraic p/t-net $N = (P, T, pre, post)$*, the* set of process terms $\mathcal{P}(N)$ *of* N *is defined inductively by the following production rules:*

$$\frac{m \in \mathbb{N}^P}{m : m \to m \in \mathcal{P}(N)}$$

$$\frac{t \in T}{t : pre(t) \to post(t) \in \mathcal{P}(N)}$$

$$\frac{\alpha_1 : m_1 \to m_1' \in \mathcal{P}(N) \land \alpha_2 : m_2 \to m_2' \in \mathcal{P}(N)}{(\alpha_1 \parallel \alpha_2) : m_1 + m_2 \to m_1' + m_2' \in \mathcal{P}(N)}$$

$$\frac{\alpha_1 : m \to m' \in \mathcal{P}(N) \land \alpha_2 : m' \to m'' \in \mathcal{P}(N)}{(\alpha_1; \alpha_2) : m \to m'' \in \mathcal{P}(N)}$$

These rules define binary operations, called concurrent composition $(\parallel)$ *and* sequential composition $(;)$ *of process terms.*

Given a process term $\alpha : m \to m'$*, we shortly say that* α *is a process term and we denote by* $pre(\alpha) = m$ *the* initial marking *and by* $post(\alpha) = m'$ final marking *of* α*.*

We define the sub-term *relation between process terms as the reflexive and transitive closure of the following relation: Given a process term $\alpha = \alpha_1; \alpha_2$ or a term $\alpha = \alpha_1 \parallel \alpha_2$, the process terms α_1 and α_2 are said to be sub-terms of α.*

Process terms are identified by an equivalence relation $\sim$ which preserves the operations $\parallel$ and ; (i.e. by a congruence w.r.t. the operations $\parallel$ and ;), given by the following axioms:
Let $m, m' \in \mathbb{N}^P$ and $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ be process terms.

(1) $(\alpha_1 \parallel \alpha_2) \sim (\alpha_2 \parallel \alpha_1)$.

(2) $((\alpha_1; \alpha_2); \alpha_3) \sim (\alpha_1; (\alpha_2; \alpha_3))$, *whenever these terms are defined.*

(3) $((\alpha_1 \parallel \alpha_2) \parallel \alpha_3) \sim (\alpha_1 \parallel (\alpha_2 \parallel \alpha_3))$.

(4) $((\alpha_1 \parallel \alpha_2); (\alpha_3 \parallel \alpha_4)) \sim ((\alpha_1; \alpha_3) \parallel (\alpha_2; \alpha_4))$, *whenever these terms are defined.*

(5) $(\alpha_1; post(\alpha_1)) \sim \alpha_1 \sim (pre(\alpha_1); \alpha_1)$.

(6) $m + m' \sim (m \parallel m')$

(7) $\alpha_1 + 0 \sim \alpha_1$ *for the empty multiset* 0.

Axiom (1) represents the commutativity of concurrent composition, axioms (2) and (3) the associativity of concurrent composition and sequential composition of process terms, axiom (4) the distributivity, axiom (5) states that elements of $\mathbb{N}^P$ are partial neutral elements with respect to ; and axiom (6) expresses that addition of two multisets is congruent to the process term constructed from their concurrent composition. The last axiom expresses the absorbtion of an empty multiset by a process term.

Observe that for any two equivalent process terms $\alpha_1 \sim \alpha_2$, we have $pre(\alpha_1) = pre(\alpha_2)$ and $post(\alpha_1) = post(\alpha_2)$.

Now, let us define inductively the number of occurrences of a transition in a process terms of an algebraic net.

Definition 55 *Let t be a transition and α be a process term of an algebraic p/t net. The number of occurrences of transitions in a process term α is a multiset $|\alpha| : T \to \mathbb{N}$ defined inductively as follows:*

- *Given $p \in P$, for the elementary process term $\alpha = p : p \to p$ define $|\alpha|$ to be the empty multiset (i.e. $|\alpha|(t) = 0$ for every $t \in T$).*
- *Given a $t \in T$, for the elementary process term $\alpha = t : m \to m'$ define $|\alpha| = t$.*
- *Given a process term $\alpha = \alpha_1 \parallel \alpha_2$ or $\alpha = \alpha_1; \alpha_2$, define $|\alpha| = |\alpha_1| + |\alpha_2|$.*

Definition 56 (Algebraic p/t-net corresponding to a p/t-net)
Let $N = (P, T, F, W)$ be a p/t-net. The algebraic p/t-net $N' = (P, T, pre, post)$ defined by $\forall p \in P, \forall t \in T : pre(t)(p) = W(p, t)$ and $post(t)(p) = W(t, p)$ is said to be corresponding *to N.*

An important role play process terms in normal forms, defined as follows:

Definition 57 (Concurrent step term, Step sequence term) *A process term generated without using the sequential composition is called* concurrent step term*.*
We define inductively step sequence terms*:*

- *A process term generated using sequential composition of concurrent step terms is called step sequence term.*
- *Sequential composition of step sequence terms is a step sequence term.*

Due to the axiom of associativity of sequential composition, we omit parentheses and write $\alpha_1; \ldots; \alpha_n$ for a step sequence term of concurrent step terms α_i, $i \in \{1, \ldots, n\}$.

Observe that directly from the definitions we obtain the following correspondence between step sequences and step sequence terms.

Proposition 58 *Let N be a p/t-net and $N' = (P, T, pre, post)$ be the algebraic p/t-net corresponding to N. A step sequence $\sigma = s_1 \ldots s_n$ is enabled to occur in m and its occurrence leads to m' if and only if there exists a step sequence term $\alpha = \alpha_1; \ldots; \alpha_n$ of N' with initial marking m and final marking m' satisfying $|\alpha_i| = s_i$ for every $i \in \{1, \ldots, n\}$.*

Special step sequence terms are maximally sequentialized process terms:

Definition 59 (Maximally sequentialized process term) *We say that a process term α_{seq} is* maximally sequentialized *iff it is of the form $(t_1 \parallel m_1); \ldots; (t_n \parallel m_n)$ with transitions $t_i \in T$ and markings $m_i \in \mathbb{N}^P$, $i \in \{1, \ldots, n\}$.*

Remark 60 *Notice that the previous proposition gives also a correspondence between occurrence sequences and maximally sequentialized terms. Consider a marked p/t-net (N, m_0) and the algebraic p/t-net $N' = (P, T, pre, post)$ corresponding to N. Observe that for every occurrence sequence $\sigma = t_1 \ldots t_n : m_0 \to m$, there are appropriate markings $m_1, \ldots, m_n$, such that $\alpha = (t_1 \parallel m_1); \ldots; (t_n \parallel m_n) : m_0 \to m$ is a defined maximally sequentialized process term of N'.*
On the other hand, for every defined process term α of the form $\alpha = (t_1 \parallel m_1); \ldots; (t_n \parallel m_n) : m_0 \to m$ with transitions $t_i \in T$ and markings $m_i \in \mathbb{N}^P$, $i = 1, \ldots, n$, the sequence of transitions $t_1 \ldots t_n$ is an occurrence sequence of N leading from m_0 to m.

Let us finish this section with a lemma stating the relationship between process terms and occurrence sequences.

Lemma 61 *Let α be a process term. Then there exists a term α_{seq} such that $\alpha \sim \alpha_{seq}$ and α_{seq} is a maximally sequentialized process term.*

Every sub-term of α of the form $\alpha_1 \parallel \alpha_2$, where α_1 and α_2 are sub-terms containing at least one transition, can be equivalently sequentialized by

$$\alpha_1 \parallel \alpha_2 \sim (\alpha_1; post(\alpha_1)) \parallel (pre(\alpha_2); \alpha_2) \sim (\alpha_1 \parallel pre(\alpha_2)); (post(\alpha_1) \parallel \alpha_2).$$

Sub-terms consisting only of sequential and concurrent compositions of markings can be (due to the equivalence rules) represented as single markings. It follows that α can be equivalently transformed into a process term, in which for all sub-terms of the form $\alpha_1 \parallel \alpha_2$ at most one of the sub-terms α_1 and α_2 contains transitions. Assume a sub-term $\alpha_1 \parallel m$ with α_1 containing a transition and m being a marking. The sub-term α_1 can be of the form $\alpha_1 = \beta_1; \beta_2$ or $\alpha_1 = \beta_1 \parallel \beta_2$. In the first case, m can be equivalently joined to β_1 and β_2 via

$$(\beta_1; \beta_2) \parallel m \sim (\beta_1; \beta_2) \parallel (m; m) \sim (\beta_1 \parallel m); (\beta_2 \parallel m).$$

In the second case, $\alpha_1 = \beta_1 \parallel \beta_2$ can be equivalently sequentialized in the same way as $\alpha_1 \parallel \alpha_2$ (observe that this recursive procedure terminates). Altogether, α can be equivalently sequentialized into a step sequence, where each concurrent step in this step sequence contains at most one transition. Applying axiom (5), one gets the searched α_{seq}.

7.2 From process terms to labelled partial orders

Throughout this section let (N, m_0) be a fixed marked p/t-net and $N' = (P, T, pre, post)$ be the algebraic p/t-net corresponding to N.

Definition 62 (Labelled partial order of a process term)
Define inductively the labelled partial order $lpo_\alpha = (V_\alpha, <_\alpha, l_\alpha)$ *of process terms α:*

- *Given a marking m, $lpo_m = (\emptyset, \emptyset, \emptyset)$.*
- *Given a transition $t \in T$, $lpo_t = (\{v\}, \emptyset, l)$, where $l(v) = t$.*
- *Given process terms α_1 and α_2 with associated labelled partial orders $lpo_1 = (V_1, <_1, l_1)$ and $lpo_2 = (V_2, <_2, l_2)$,*

  $$lpo_{\alpha_1 \parallel \alpha_2} = (V_1 \cup V_2, <_1 \cup <_2, l_1 \cup l_2),$$

  where the sets of nodes V_1 and V_2 are assumed to be disjoint (what can be achieved by appropriate renaming of nodes).

- *Given process terms* α_1 *and* α_2 *with associated labelled partial orders* $lpo_1 = (V_1, <_1, l_1)$ *and* $lpo_2 = (V_2, <_2, l_2)$*,*

$$lpo_{\alpha_1;\alpha_2} = (V_1 \cup V_2, <_1 \cup <_2 \cup (V_1 \times V_2), l_1 \cup l_2),$$

where the sets of nodes V_1 *and* V_2 *are assumed to be disjoint (what can be achieved by appropriate renaming of nodes).*

Obviously, the previous definition is sound in the sense, that the structures attached to process terms are labelled partial orders. Notice that there are several labelled partial orders of a process term, but all these labelled partial orders have up to renaming the same structure, i.e. they are isomorphic.

In the introduction we have mentioned that the labelled partial orders of process terms of p/t nets can be characterized by their shape.

Definition 63 *A labelled partial order* $lpo = (V, <, l)$ *is called N-free if it fulfils:* $\forall a, b, c, d \in V : a < c \wedge b < d \wedge a < d \wedge a \, co \, b \wedge c \, co \, d \Rightarrow b < c$.

*Let us notice, that the LPOs of process terms of p/t nets coincide with so called finite series-parallel pomsets, i.e. with LPOs generated from single element LPOs by concurrent composition (disjoint union side by side) and sequential composition. It is a well known fact (see e.g. [***Gischer***]), that a finite labelled partial order is series-parallel if and only if it is N-free. As a consequence we get:*

Theorem 64 *A labelled partial order is N-free if and only if it is a labelled partial order of a process term of a p/t net.*

Remark 65 *Observe that the only axiom which changes the ordering of events is distributivity: if* $(V, <_\alpha, l)$ *is a labelled partial order of a process term* α *and another process term* β *is equivalent to* α *through the axioms* $(1)-(3)$ *and* $(5)-(7)$*, then* $(V, <_\alpha, l)$ *is also a labelled partial order of* β*. Thus, we will not distinguish between process terms equivalent according to the axioms except distributivity.*

In the case of distributivity, for a process term $\alpha = (\alpha_1; \alpha_3) \parallel (\alpha_2; \alpha_4)$ *with a labelled partial order* $(V, <_\alpha, l)$ *and the equivalent process term* $\beta = (\alpha_1 \parallel \alpha_2); (\alpha_3 \parallel \alpha_4)$ *there exists a labelled partial order* $(V, <_\beta, l)$ *of* β *satisfying* $<_\alpha \subset <_\beta$*. Namely, in* $<_\beta$ *orderings between events in* α_1 *and* α_4 *as well as* α_2 *and* α_3 *are added w.r.t.* $<_{\alpha_2}$*.*

Directly from the related definitions we obtain the following proposition, which completes the relationship between different descriptions of occurrence sequences.

Proposition 66 *Let* N *be a p/t-net and* $N' = (P, T, pre, post)$ *be the algebraic p/t-net corresponding to* N*. The labelled partial order* $lpo = (V, <, l)$ *is associated to an occurrence* *sequence of N $\sigma = t_1 \dots t_n$ enabled to occur in m and leading to m' if and only if it is the labelled partial order of a maximally sequentialized process term $\alpha = (t_1 \parallel m_1); \dots ; (t_n \parallel m_n)$ of N' with initial marking m and final marking m' satisfying $m_i \in \mathbb{N}^P$ for every $i \in \{1, \dots, n\}$.*

From the construction of labelled partial orders of process terms, Lemma 61 and the previous proposition we get:

Lemma 67 *Let α be a process term. Then there exists a maximally sequentialized process term α_{seq} such that $\alpha_{seq} \sim \alpha$ and the labelled partial order of α_{seq} is a linearization of the labelled partial order of α.*

Similarly to the Proposition 66, the following proposition completes the relationships between different descriptions of step sequences.

Proposition 68 *Let N be a p/t-net and $N' = (P, T, pre, post)$ be the algebraic p/t-net corresponding to N. A labelled partial order $lpo = (V, <, l)$ is associated to a step sequence $\sigma = s_1 \dots s_n$ of N enabled to occur in m and leading to m' if and only if it is the labelled partial order of a step sequence term $\alpha = \alpha_1; \dots ; \alpha_n$ of N' with initial marking m and final marking m' satisfying $|\alpha_i| = s_i$ for every $i \in \{1, \dots, n\}$.*

This relationship has the following consequence:

Proposition 69 *A labelled partial order $lpo = (V, \prec, l)$ is enabled to occur in a marking m w.r.t N, if and only if for every slice S of $(V, \prec)$ there exists a step sequence term β of N' with initial marking m which has the labelled partial order $(V, <_\beta, l)$ satisfying: S is a slice of $(V, <_\beta)$ and $\prec \subseteq <_\beta$.*

Because we want to intersect and compare the labelled partial orders, it is convenient to identify copies of events of a process term with nodes of the associated labelled partial order lpo_α. Formally, given a set of events V together with a labelling function $l : V \to T$, we define a (V, l)-copy net of the algebraic p/t-net N' as a new algebraic p/t-net with transitions given by events from V. The source and the target function for events from V are determined by the labelling function. A suitable extension of the labelling function will map process terms of the V-copy net to process terms of the original net.

Definition 70 ((V, l)-copy net of an algebraic p/t-net) *Let V be set and $l : V \to T$ be a labelling function. Denote by $N_{(V,l)}$ the algebraic p/t-net $N_{(V,l)} = (P, V, pre_{(V,l)}, post_{(V,l)})$, where $pre_{(V,l)}(v) = pre(l(v))$ and $post_{(V,l)}(v) = post(l(v))$ for every $v \in V$. $N_{(V,l)}$ is called* (V, l)-copy net of *N'.*

We extend inductively the labelling l for process terms of $N_{(V,l)}$ as follows:

- *Given a marking* m, $l(m) = m$.
- *Given process terms* α_1 *and* α_2 *of* $N_{(V,l)}$, $l(\alpha_1 \parallel \alpha_2) = l(\alpha_1) \parallel l(\alpha_2)$.
- *Given process terms* α_1 *and* α_2 *of* $N_{(V,l)}$ *such that* $\alpha_1;\alpha_2$ *is defined,* $l(\alpha_1;\alpha_2) = l(\alpha_1);l(\alpha_2)$.

Remark 71 *Because events from* V *labelled on a transition* t *are only its copies (have the same source and target), it is easy to observe that the extended labelling* l *preserves the following properties, each of which can be proven by induction:*

- *It preserves initial and final marking of process terms, i.e.* $pre(\alpha) = pre(l(\alpha))$ *and* $post(\alpha) = post(l(\alpha))$ *for every process term* α *of* $N_{(V,l)}$.
- *It maps process terms of* $N_{(V,l)}$ *to process terms of* N'.
- *It preserves the* $\sim$*-equivalence of process terms.*
- *It is surjective, i.e. for every process term* α' *of* N' *there is a process term* α *of* $N_{(V,l)}$ *with* $l(\alpha) = \alpha'$.
- *It preserves the partial order of transitions, i.e. if* $(V, <, id)$ *is a labelled partial order of a process term* α *of* $N_{(V,l)}$, *then* $lpo' = (V, <, l)$ *is a labelled partial order of* $l(\alpha)$.
- *For each process term* α' *of* N' *with labelled partial order* $lpo' = (V, <, l)$, *there is a process term* α *of* $N_{(V,l)}$ *with* $l(\alpha) = \alpha'$ *and with labelled partial order* $lpo = (V, <, id)$.
- *If* $(V, <, id)$ *is a labelled partial order of a process term* α *of* $N_{(V,l)}$ *then* $<$ *is unique, i.e. there exists no partial order* $<'$ *on* V *such that* $< \neq <'$ *and* $(V, <', id)$ *is a labelled partial order of* α.

In order to simplify the identification of labelled partial orders and process terms, we will use process terms of the (V, l)*-copy net which enable the unique identification.*

Definition 72 (Copy term) *Let* $lpo = (V, <, l)$ *be a labelled partial order of a process term* α *of* N'. *Then the unique process term* α^{lpo} *of the* (V, l)*-copy net* $N_{(V,l)}$ *such that* $(V, <, id)$ *is a labelled partial order of* α^{lpo} *and* $l(\alpha^{lpo}) = \alpha$ *is called the copy term of* α *w.r.t.* lpo.

In the introduction to this chapter we promise to show that a labelled partial order of a process term is enabled to occur in its initial marking. It follows from the following lemma.

Lemma 73 *Let α be a process term of N' with initial marking m, let $(V,<,l)$ be the labelled partial order of α, and let S be a slice of $(V,<)$. Then there exists a step sequence term β_V of the (V,l)-copy of N' with initial marking m, which has the labelled partial order $(V,<_\beta,id)$ satisfying:*

- $\beta_V \sim \alpha_{(V,l)}$
- *S is a slice of $(V,<_\beta)$,*
- $< \subseteq <_\beta$

By the last remark, there exists a process term α_V of $N_{(V,l)}$ such that:

- *α_V contains each $v \in V$,*
- *$l(\alpha_V) = \alpha$,*
- *$lpo_V = (V,<,id)$ is a labelled partial order of α_V.*

We will equivalently transform α_V into a step sequence term β_V of the form $\beta_V = \beta_1;\beta_S;\beta_2$, where β_1 and β_2 are in maximally sequentialized form and β_S is a step sequence containing exactly the events of S, which then has the searched properties.

Let α_S^1 be the smallest sub-term of α_V, which contains all $v \in S$. Similarly as in the proof of Lemma 61, we can equivalently sequentialize the rest of the process term α_V to get a process term α_V^1 of the form

$$\alpha_V^1 = \alpha_1;\alpha_S^1 \parallel m;\alpha_2,$$

where α_1 and α_2 are in maximally sequentialized form and m is a marking. By this equivalence transformation, some ordering is added, but no ordering between events of S. That means there is a labelled partial order $(V,<^1,id)$ of α_V^1 such that $< \subseteq <^1$ and S is a slice of $<^1$. If α_S^1 is a concurrent step, then $l(\alpha_V^1)$ is a step sequence and $(V,<^1,l)$ is a labelled partial of $l(\alpha_V^1)$ with slice S. In this case, $\beta_V = \alpha_V^1$ is the searched process term.

If α_S^1 is not a concurrent step term, we further equivalently transform α_S^1 according to the following procedure:

Because α_S^1 is the smallest sub-term of α_V containing all events from S and S is a co-set in $(V,<^1)$, it is of the form $\alpha_S^1 = \gamma_1' \parallel \gamma_2'$, where at least one of the sub-terms γ_1',γ_2' contains a sub-term of the form $\delta_1';\delta_2'$.

Thus, α_S^1 can be equivalently transformed, only using the axioms of commutativity and associativity of $\parallel$ (without changing the ordering of events), into

$$\alpha_S^{1'} = \gamma \parallel (\delta_1;\delta_2).$$

Since S is a maximal co-set, both sub-terms γ and $\delta_1;\delta_2$ contain events from S. Without loss of generality we assume that neither δ_1 nor δ_2 equal a marking, since then $\alpha_S^{1'}$ can be equivalently transformed into a concurrent step term and we are done. That means $\alpha_S^{1'}$ satisfies: γ contains some events from S and either

(a) δ_1 contains some events from S and δ_2 contains some events from V but no events from S, or

(b) δ_1 contains no events from S but some other events from V and δ_2 some events from S.

In case (a), we have

$$\gamma \parallel (\delta_1;\delta_2) \sim (\gamma; post(\gamma) \parallel (\delta_1;\delta_2) \sim (\gamma \parallel \delta_1); (post(\gamma) \parallel \delta_2).$$

This transformation removes no ordering between events of V. It just adds ordering between events of γ and δ_2, but adds no ordering between events of S (S remains a slice). Thus, after a maximal sequentialization of $post(\beta) \parallel \delta_2$ into a process term α_3, we get

$$\alpha_V^1 \sim \alpha_V^2 = \alpha_1; ((\gamma \parallel \delta_1) \parallel m); \alpha_3; \alpha_2.$$

By construction, there is a labelled partial order $(V, <^2, id)$ of α_V^2 with $< \subseteq <^2$ containing the slice S. Now, because S is a maximal set of unordered events, the smallest sub-term of α_V^2 containing all events of S is $\alpha_S^2 = \gamma \parallel \delta_1$. In case (b) we get a similar transformation.

If α_S^2 is a concurrent step term, then we are done as above. If not, we repeat the procedure, in each step reducing the number of events in the smallest sub-term containing all events of S. Thus, after finitety many steps we get a process term α_V^n such that:

- *$\alpha_V^n = \alpha_1^n; \alpha_S^n; \alpha_2^n$ where α_1^n, α_2^n are in maximally sequentialized form and α_S^n contains only events from S.*
- *There is a labelled partial order $(V, <^n, id)$, in which S is a slice, such that $<^n \supseteq <$.*

This implies that α_S^n has to be a concurrent step term. This finishes the proof.

Theorem 74 *A labelled partial order $(V, \prec, l)$ of any process term of N' is enabled to occur in m w.r.t. N.*

In Section 5.2 we have defined compatible labelled partial orders as labelled partial orders, intersections of which are enabled labelled partial orders. Now, let us shift the definition of compatibility to (copy) process terms.

Definition 75 (Compatible set) *Let $N = (P, T, F, W)$ be a p/t net, m be a marking of N and N' be the algebraic p/t net associated to N. Let V be a set and $l : V \to T$ be* *a labelling. Then the set* Υ *of process terms of* $N_{(V,l)}$ *with initial marking* m *is called compatible w.r.t.* N *and* m *iff each process term* α *from* Υ *has the labelled partial order of the form* $lpo_\alpha = (V, <_{alpha}, id)$ *and the set of labelled partial orders* $X = \{(V, <_\alpha, l) \mid \alpha \in \Upsilon\}$ *is compatible w.r.t.* N *and* m*. The labelled partial order associated to* X *is said to be associated to* Υ*.*

Obviously, according to the correspondence between step sequences and step sequence terms we have the following proposition:

Proposition 76 *Let* $lpo = (V, \prec, l)$ *be a labelled partial order. Then* lpo *is enabled to occur in* m *w.r.t.* N *if and only if there exists a set* Υ *of (copy) process terms compatible w.r.t.* N *and* m *such that* lpo *is associated to* Υ*.*

In other words, the previous proposition says that every enabled labelled partial order can be constructed *from LPOs associated to process terms.*

7.3 Summary revisited

In this section we summarize the process term semantics of p/t nets w.r.t. the partial order based semantics of p/t nets.

Denote by **TERM** *the (isomorphism classes of) LPOs associated to process terms of an algebraic p/t net* N' *with initial marking* m_0 *we get:*

$$\textbf{STEPSEQ} \subseteq \textbf{TERM} \subseteq \textbf{ENABLED}$$

$$\textbf{TERM} \Supset \textbf{ENABLED}$$

Labelled partial orders representing different semantics of a p/t net N *with an initial marking* m_0 *can be obtained from process terms of the corresponding algebraic p/t net* N' *as follows:*

- **SEQ** *equals the set of (isomorphism classes of) LPOs of maximally sequentialized process terms with initial marking* m_0*.*
- **STEPSEQ** *equals the set of (isomorphism classes of) LPOs of step sequence terms with initial marking* m_0*.*
- **ENABLED** *equals the set of (isomorphism classes of) LPOs associated with compatible sets of (copy) process terms with initial marking* m_0*.*
- **MINRUN** *equals the set of minimal LPOs from the set* **ENABLED***, i.e. the set* **MINENABLED***.*

7.4 From process terms to commutative runs

*In order to finish the overall picture of relationship between process terms and different partial order based semantics of p/t nets, let us discuss the result of [***DMM96***] about the one-to-one correspondence of process term equivalence classes and commutative processes.*

A similar result can be obtained from the relationship between process terms, occurrence sequences and commutative processes using the following lemma.

Lemma 77 *Let N be a p/t net, m be a marking of N and N' an algebraic p/t net corresponding to N. Let σ, τ be occurrence sequences of N enabled to occur in m and let α, β be maximally sequentialized process term of N' with initial marking m satisfying: the labelled partial order of α is associated to σ and the labelled partial order of β is associated to τ. Then $\sigma \equiv_0^* \tau$ if and only if $\alpha \sim \beta$.*

In the first part of the proof we will show $\sigma \equiv_0^* \tau$ implies $\alpha \sim \beta$. It suffices to show that $\sigma \equiv_0 \tau$ implies $\alpha \sim \beta$. Let $\sigma \equiv_0 \tau$. Then we have $\sigma = t_1 \dots t_{i-1} t_i t_{i+1} t_{i+2} \dots t_n, \tau = t_1 \dots t_{i-1} t_{i+1} t_i t_{i+2} \dots t_n$, and $\varphi = \{t_1\} \dots \{t_{i-1}\}\{t_i, t_{i+1}\}\{t_{i+2}\} \dots \{t_n\}$ is a step sequence of N enabled to occur in m_0.

According to the relationship between occurrence sequences, step sequences and process terms, we also have that $\alpha = (t_1 \parallel m_1); \dots ; (t_{i-1} \parallel m_{i-1}); (t_i \parallel m_i); (t_{i+1} \parallel m_{i+1}); (t_{i+2} \parallel m_{i+2}); \dots ; (t_n \parallel m_n)$ and $\beta = (t_1 \parallel m_1); \dots ; (t_{i-1} \parallel m_{i-1}); (t_{i+1} \parallel m'_{i+1}); (t_i \parallel m'_i); (t_{i+2} \parallel m_{i+2}); \dots ; (t_n \parallel m_n)$ are maximally sequentialized process terms of N' with initial marking m and $\gamma = (t_1 \parallel m_1); \dots ; (t_{i-1} \parallel m_{i-1}); (t_i \parallel t_{i+1} \parallel m''_i); (t_{i+2} \parallel m_{i+2}); \dots ; (t_n \parallel m_n)$ is a step term of N' with initial marking m. Because $\sim$ is a congruence on process terms w.r.t. sequential composition, it is enough to show that the terms $\alpha' = (t_i \parallel m_i); (t_{i+1} \parallel m_{i+1})$ and $\beta' = (t_{i+1} \parallel m'_{i+1}); (t_i \parallel m'_i)$, which have the same initial marking m' and the same final marking m'', are $\sim$-equivalent. This can be easily done using the term $\gamma' = (t_i \parallel t_{i+1} \parallel m''_i)$ with initial marking m' and final marking m''. Thus, we have $m' = pre(t_i) + pre(t_{i+1} + m''_i)$ and therefore $m_i = pre(t_{i+1}) + m''_i$. Similarly, $m'' = post(t_i) + post(t_{i+1} + m''_i)$ and therefore $m_{i+1} = post(t_i) + m''_i$. From this we have $\gamma' = (t_i \parallel t_{i+1} \parallel m''_i) \sim (t_i; post(t_i)) \parallel (pre(t_{i+1} \parallel m''_i); (t_{i+1} \parallel m''_i)) \sim (t_i \parallel pre(t_{i+1} \parallel m''_i)); (post(t_i) \parallel (t_{i+1} \parallel m''_i)) \sim (t_i \parallel m_i); (t_{i+1} \parallel m_{i+1}) = \alpha'$. Analogously, $\gamma' \sim \beta'$, what finishes the first part of the proof.

In the second part of the proof we will show that $\alpha \sim \beta$ implies $\sigma \equiv_0^* \tau$. The fact that $\alpha \sim \beta$ means that there is a finite sequence of transformations given by axioms $1 - 7$. According to Remark 65, the only axiom which changes the labelled partial order of terms is distributivity: $\beta = ((\alpha_1 \parallel \alpha_2); (\alpha_3 \parallel \alpha_4)) \sim ((\alpha_1; \alpha_3) \parallel (\alpha_2; \alpha_4)) = \alpha$. Using this axiom the LPO of β is obtained from the LPO of α by adding some ordering, but deleting no ordering. Because α, β are maximally sequentialized process terms, we can split any equivalence transformation $\alpha \sim \beta$ into a finite sequence of alternating transformations[1]

[1] where each of these transformations can consists of several applications of axioms 1-7

$\alpha = \gamma_1 \sim \gamma_2 \sim \ldots \gamma_n = \beta$ in such a way, that for the LPO $lpo_{i+1}(V, <_{i+1}, l)$ of γ_{i+1} and for the LPO $lpo_i(V, <_i, l)$ of γ_i holds:

- *$<_{i+1} \subseteq <_i$, i.e. some ordering is eventually deleted but no ordering is added for all odd i such that $0 < i < n$*
- *$<_i \subseteq <_{i+1}$, i.e. some ordering is eventually added, but no ordering is deleted for all even i such that $0 < i < n$*

and n is an odd number. In general, some terms γ_i with odd index are not necessarily maximally sequentialized. However, according to Lemma 67 we can sequentialize any such γ_i into an equivalent maximally sequentialized γ_i' by just adding some ordering to its LPO. Then we have $\gamma_{i-1} \sim \gamma_i' \sim \gamma_{i+1}$ where the LPO of γ_i' is obtained from the LPO of γ_{i-1} by adding some ordering and the LPO of γ_{i+1} is obtained from the LPO of γ_i by deleting some ordering. Therefore, whenever γ_i with an odd index is not maximally sequentialized, we can replace it with a maximally sequentialized process term. Thus, we can consider without loss of generality γ_i for all odd indexes to be maximally sequentialized process terms. Denote by μ_i the occurrence sequence with the LPO of γ_i for all odd i.

Now, fix an even index i. According to Theorem 74 we have that the LPO lpo_i of γ_i is enabled. From Theorem 28 it is a sequentialization of a run. Because the LPOs of γ_{i-1} and γ_{i+1} are linearizations of lpo_i, they are also linearizations of the run. According to Theorem 49, $\mu_{i-1} \equiv_0^* \mu_{i+1}$. It follows $\sigma = \mu_1 \equiv_0^* \mu_n = \tau$.

Theorem 78 *Let N be a p/t net, m_0 be a marking of N and N' be the algebraic p/t net corresponding to N. Let α^1, α^2 be process terms of N' with initial marking m_0 and K_1, K_2 be processes of (N, m_0) such that the LPOs of α^1, α^2 are sequentializations of runs representing K_1, K_2, respectively. Then $K_1 \equiv_1^* K_2$ if and only if $\alpha^1 \sim \alpha^2$.*

According to Lemma 67 we can sequentialize α^i ($i \in \{1, 2\}$) into a maximally sequentialized process term α^i_{seq} satisfying: $\alpha^i_{seq} \sim \alpha^i$ and the LPO of α^i_{seq} is a sequentialization of α^i and therefore also a sequentialization of the run representing K_i.

Then from $K_1 \equiv_1^* K_2$ we get according to Theorem 49 that the occurrence sequences associated with the LPOs of α^i_{seq} are exchange equivalent. This implies, according to the last lemma, that $\alpha^i_{seq} \sim \alpha^i_{seq}$. It follows that $\alpha^1 \sim \alpha^2$.

If $\alpha^1 \sim \alpha^2$, then $\alpha^1_{seq} \sim \alpha^2_{seq}$. From the last lemma, the occurrence sequences associated with the LPOs of α^i_{seq} are exchange equivalent. From Theorem 49 $K_1 \equiv_1^* K_2$.

From the previous results we also get the following corollary.

Corollary 79 *Let Υ be a compatible set of (copy) process terms w.r.t. N and m_0. Then any two (copy) process terms from Υ are $\sim$ equivalent.*

Chapter 8

Conclusion

*In this part, we have summarized results on relationship between different semantics of p/t nets based on labelled partial orders. Then, we have discussed relationships between process terms of place/transition nets over a simple algebra used in [**MM90**; **DMM96**] for "collective token semantics" and different variants of semantics of p/t nets based on (labelled) partial orders. Namely, we have attached labelled partial orders to single process terms and investigated their relationship to:*

- *occurrence sequences,*
- *step sequences,*
- *enabled labelled partial orders, and*
- *runs defined by single processes defined in [**GR**], which determine "individual token semantics".*

*Last but not least, we have discussed the relationship between equivalence classes of process terms and the swapping equivalence classes of processes (i.e. commutative processes), which determine "collective token semantics". To show this correspondence using LPOs attached to single process terms makes it much easier than similar result of [**DMM96**], or direct proof of this result presented in [**OS03**].*

*In particular, we provided a simple way to obtain all "individual" minimal runs from process terms for a given place/transition nets. In other words, the presented results illustrate that process term semantics (even over a simple algebra of free commutative monoids) can be used to derive all reasonable partial order based semantics of p/t nets, including "individual token semantics", which could up to now be derived only using more sophisticated algebras ([**Sassone98**; **SassoneIntro**]).*

Describe theoretical framework, algorithms, models, or experimental design. Include equations where relevant, e.g.,

$$J(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\ell\big(f_\theta(x), y\big)\big] + \lambda\,\|\theta\|_2^2. \tag{8.1}$$

8.1 Data

Data sources, collection, preprocessing, and quality checks. Mention ethics and privacy considerations if applicable.

8.2 Implementation Details

Software stack, hardware, versions, and reproducibility notes. Include code snippets using `listings`:

```
for epoch in range(epochs):
    for batch in loader:
        loss = model.update(batch)
        if step % 100 == 0:
            print(step, loss.item())
```

Listing 8.1: Example training loop

Chapter 9

Results to Date

Summarise completed experiments/analyses. Use figures and tables with clear captions.

9.1 Quantitative Results

Table 9.1: Interim performance on validation set

Model	Accuracy (%)	Precision (%)	Recall (%)
Baseline A	78.3	75.1	76.2
Prototype B	82.7	81.4	80.2

9.2 Qualitative Results

Add representative examples, error analyses, ablation notes, or case studies.

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VAIA VÝSKUMNÁ AGENTÚRA

Chapter 10

Discussion

Interpret interim results, limitations, threats to validity, and implications. Compare against related work.

Chapter 11

Work Plan

11.1 Completed Milestones

Bullet list with dates.

11.2 Upcoming Milestones and Timeline

Outline remaining tasks with target dates. A simple timeline table:

Table 11.1: Planned timeline (next quarter)

Start	End	Task
2025-10-01	2025-10-15	Data augmentation experiments
2025-10-16	2025-11-05	Model tuning and validation
2025-11-06	2025-11-20	Error analysis and ablations
2025-11-21	2025-12-05	Write-up of final report

11.3 Risks and Mitigations

Identify top risks (technical, data, resourcing) and planned mitigations.

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD VLÁDY SLOVENSKEJ REPUBLIKY VAIA VÝSKUMNÁ AGENTÚRA

Chapter 12

Resources and Budget (Optional)

Summarise compute, software, datasets, and estimated costs.

Chapter 13

Ethics, Privacy, and Data Management

Describe data handling, consent, privacy safeguards, bias assessment, and data management plan.

Chapter 14

Conclusion

Recap progress, highlight contributions so far, and restate next steps.

Appendix A

Supplementary Material

Additional figures, tables, or proofs.

Appendix B

Glossary and Acronyms (Optional)

Define specialised terms and acronyms used in the report.