1 Introduction

1.1 The type of the thesis and the specific subfield

This thesis is a research and software development project. It falls within the field of Petri net theory and compilers, with applications in modeling and simulation of concurrent systems.

1.2 General presentation of the topic

Petri nets are a mathematical modeling tool used to describe and analyze concurrent, distributed, and parallel systems. They provide a formal way to represent states and transitions, making them useful in verifying properties such as deadlocks, reachability, and synchronization in complex systems.

Compilers, on the other hand, are essential in transforming high-level representations into executable models, automating the process of simulation and analysis. By combining Petri net theory with compiler techniques, we can create a language that enables users to define Petri nets in a structured manner and automatically generate executable code for simulation. This approach facilitates the modeling, testing and visualization of concurrent systems, making it easier to analyze their behavior.

1.3 Purpose and motivation for choosing the topic

Modeling concurrent and parallel systems is essential for analyzing distributed processes, formal verification, and their optimization. Petri nets are a powerful tool in this field, and a dedicated language can simplify the definition and testing of models. The motivation for choosing this topic is both the theoretical interest in formal systems and the practical applicability in verifying concurrent software.

1.4 Own contribution to the thesis

I have developed a description language for Petri nets and a compiler written in Haskell that transforms this language into a GoLang executable representation and LaTeX code for visualization. The project combines concepts from formal language theory, compilers, and concurrent system modeling.

1.5 Structure of the thesis

1. Introduction: Presents the context of the thesis, motivation, and personal contribution.

2. TBA

2 Preliminaries

2.1 Scientific and Technological Concepts Underlying the Topic

Definition 1. *A **net** is a triple $N = (P, T, F)$ where: [1]*

1. *P and T are finite disjoint sets of **places** and **transitions**.*

2. *$F \subseteq (P \times T) \cup (T \times P)$ is a set of flow relations.*

3. *for every $t \in T$ there exists $p, q \in P$ such that $(p, t), (t, q) \in F$.*

4. *for every $t \in T$ and $p, q \in P$, if $(p, t), (t, q) \in F$, then $p \neq q$.*

Having a net $N = (P, T, F)$ and $t \in T$, we note $\bullet t$ the incoming arcs into t and $t\bullet$ the outcoming arcs from t. The notation is analogous for $p \in P$.

Definition 2. *A **Petri net** is a net of the form $PN = (N, M, W)$, where: [1]*

1. *$N = (P, T, F)$ is a net.*

2. $M : P \to Z$ *is a place multiset, where* Z *is a countable set.* M *maps for every place the number of tokens it has, and it is known as the* ***initial marking****, often noted with* $M0$ *or* $m0$*.*

3. $W : F \to Z$ *is an arc multiset, which denotes the arc's weight.*

Definition 3. *Having a Petri net* $PN = (N, M, W)$ *and* $N = (P, T, F)$*: [2]*

1. *A transition* $t \in T$ *can be* ***fired*** *if for every arc* $a = (p, t) \in \bullet t$*,* $W(a) \leq M(p)$*.*

2. *By firing t, a new marking M' is created like this:*

$$M' = \{(p, x)|p \in P, x = M(p) - W((p, t)) + W((t, p))\}$$

3. *If* $(p, t) \notin T$ *or* $(t, p) \notin T$*, then* $W(p, t)$*, respectively* $W(t, p)$ *will be subtituted with 0, or its equivalent in Z.*

4. *If* M' *can be obtained by firing a transition from* M*, then this relation is noted* $M \to M'$*.*

5. *If* M' *can be obtained by firing an arbitrary number of transitions starting with* M *then* M' *is reachable from* M*, and reachability is noted like this:* $M \xrightarrow{*} M'$*.*

Cicle detection Timed petri nets Coloured petri nets Inhibitor Arcs Highlevel nets

2.2 Current State of the Specific Subfield

********TBA

2.3 Objectives of the Thesis in Context

The main goal of this thesis is to develop a domain-specific language for defining Petri nets, along with a compiler that translates these definitions into executable code for simulation and LaTeX code for visualization. The specific objectives are:

- To design a simple and expressive language for defining Petri nets.

- To implement a compiler in Haskell that generates both simulation-ready code and graphical representations.

- To compare this approach with existing tools and evaluate its advantages in terms of usability, automation, and flexibility.

This project aims to bridge the gap between formal modeling techniques and practical implementation, making Petri net analysis more accessible and automated.

References

[1] G. Rozenberg and J. Engelfriet, “Elementary net systems,” in *Advanced Course on Petri Nets*. Springer, 1996, pp. 12–121.

[2] M. Diaz, *Petri nets: fundamental models, verification and applications*. John Wiley & Sons, 2013.