THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale : Information, Structures, Systèmes

Unité de recherche LIRMM

Vérification d'une méthodologie pour la conception de systèmes numériques critiques

Présenté par Vincent IAMPIETRO

Le Date de la soutenance

Sous la direction de David Delahaye
et David Andreu

Devant le jury composé de

[Nom Prénom], [Titre], [Labo]	[Statut jury]
[Nom Prénom], [Titre], [Labo]	[Statut jury]
[Nom Prénom], [Titre], [Labo]	[Statut jury]

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

SITPN	**S**ynchronously executed **I**nterpreted **T**ime **P**etri **N**et with priorities
VHDL	**V**ery high speed integrated circuit **H**ardware **D**escription **L**anguage
CIS	**C**omponent **I**nstantiation **S**tatement
PCI	**P**lace **C**omponent **I**nstance
TCI	**T**ransition **C**omponent **I**nstance
GPL	**G**eneric **P**rogramming **L**anguage
HDL	**H**ardware **D**escription **L**anguage
LRM	**L**anguage **R**eference **M**anual
DSL	**D**omain **S**pecific **L**anguage
MDE	**M**odel-**D**riven **E**ngineering

For/Dedicated to/To my...

Chapter 1

Preliminary Notions

In this chapter, we introduce the mathematical formalisms and notations used throughout this thesis to express and formalize our ideas. Also, in the last section, we provide the basics to understand the Coq proof assistant, which is the system we use to write our programs and to mechanize our proofs. This chapter is inspired by the book *The Formal Semantics of Programming Languages: an Introduction*, the courses of the Cambridge of formal semantics, the documentation of the Coq proof assistant, and the Coq programming with dependent-types by Adam Chlipala.

add r

1.1 Set theory and mathematical notations

1.1.1 Intuitionistic first order logic

The intuitionistic first order logic constitutes our framework for the expression and the interpretation of logical formulas. The language to express logical formulas is the same between classical and intuitionistic first order logic. A logical formula is either:

- a predicate (i.e. an atomic formula). A predicate P possibly takes n parameters as inputs and is interpreted to either true, represented by the $\top$ symbol, or false, represented by the $\bot$ symbol. We write $P(x_0, \ldots, x_n)$ to denote an n-ary predicate.
- the composition of subformulas with one of the following connectors: the conjunction $\wedge$, the disjunction $\vee$, the implication $\Rightarrow$, the double implication $\Leftrightarrow$
- a subformula prefixed by the universal quantifier $\forall$ or the existantial quantifier $\exists$. For instance, the formula $\forall x.P(x)$ denotes the atomic formula $P(x)$ where x is a unversally quantified variable of the formula. As a shorthand notation, we write $\forall x, y, z. \ldots$ to denote $\forall x, \forall y, \forall z. \ldots$. The same stands for the existantial quantifier $\exists$.

The difference between the classical first order logic and the intuitionistic one relies in the absence of the *law of the excluded middle* in the latter logic. The *law of the excluded middle* considers that for all n-ary predicate $P \in X_0$ where $x_0 \in X_0, \ldots, x_n \in X_n$, either $P(x_0, \ldots, x_n)$ is valuated to true, or

1.1.2 Set theory

In this thesis, we use set theory as the base formalism for all our mathematical definitions and proofs. In the set theory, a set represents a group of elements called the members of the set. For every set X, we write $x \in X$ to denote that the element x is a member of set X. From here, there are multiple ways to define a set.