# Chapter 1

# Introduction

## 1.1 Motivation

In some practical applications of Computer Science, the well-known Normal Forms CNF (conjunctive normal form) and DNF (disjunctive normal form) do not provide a natural framework to represent knowledge and to reason. In fact, performing inferences efficiently with formulas whose forms are non restricted to the classical ones is a matter of major interest in many practical applications inside very heterogeneous areas such as Expert Systems, Deductive Data Bases, Hardware Design, Automated Software Verification, Symbolic Optimization, Logic Programming, Automated Theorem Proving, Petri Nets, Truth Maintenance Systems, etc.

However, most of the existing efficient proof methods are designed to work with CNF formulas. So, it is a common practice to translate knowledge representations from general forms to CNF's [35, 92]. This transformation was originally proposed in 1970 by Tseitin [96] who published the first algorithm. Now, it is known that any propositional formula can be translated to another equivalent formula in the CNF form applying only the Morgan's rules or other optimized algorithms [92]. The resulting formula is no necessarily unique except if the connectives in the original formula are only $\neg$, $\wedge$ and the or-exclusive connective [64]. The translation process occurs in polynomial time if some auxiliary propositions are allowed in the CNF formula, but it takes exponential time if they are not allowed [47]. A discussion of the advantages of the CNF formulas in the context of theorem proving can be found in [86, 84] In the case of other logics, Henschen et all [61] included the case for first-order logic, Mints [80] covered the cases for modal and intuitionistic logics and finally, Hähnle [57] investigated the problem of translating arbitrary finitely valued logics to short CNF signed formulas.

Currently, two transformations are applied, one preserve the logical equivalence and the other only the satisfiability equivalence.

1. In the first case, the translation cannot skip the explosion of the number

of symbols due to the $\wedge/\vee$ distribution operation and thus the size of the resulting CNF formula can increase exponentially.

2. The other approach consists in modifying the formula by introducing artificial literals [96] aiming at preserving the satisfiability relation. This second line of solution has two strong drawbacks. First, the logical equivalence relation is lost which could be invalid for certain applications. Second, to solve the SAT problem two procedures are required: the first one transforms the original formula into a CNF formula, and the second one, taking as its input the translated CNF formula which is bigger than the original one, applies properly the satisfiability test.

Hence, processing directly the non-clausal formula in an appropriated way arises as the most efficient approach of solving non-clausal SAT problems.

## 1.2 Contributions

On the theoretical side, our contribution described here aims at pushing further the frontiers of non-clausal tractability. Thus, we firstly have defined a new class of formulas in Negation Normal Form having a Horn-like shape. In this sense, the proposed formulas absorb the Horn language as a particular case. Secondly, we have established a set of inference rules which are sound and refutationally complete. In third place, we have designed respectively, strictly linear algorithms to solve the propositional satisfiability problem and almost linear algorithms to solve the many-valued satisfiability problem.

On the practical side, as the formulas keep a Horn-like structure, they are of relevant interest in such applications as for instance those based in Rule Based Systems. Indeed, the rules and the questions of many real applications require to represent and reason with a richer language than the Horn formulas language. The proposed formulas represent logically equivalent pure Horn problems but with exponentially less symbols. Hence, as the described algorithm runs in linear time on this class, the gain of time can be of an exponential order with respect to the known linear algorithms running on the Horn formulas.

## 1.3 Structure of the thesis

The thesis is organized in six chapters, whose contents are summarized below:

**Chapter 1. Introduction**
In this chapter, we describe first the motivations which lead us to work on this research. Second, we mention briefly the importance we believe our results and scientific contributions have. Finally the structure of this thesis is described.

**Chapter 2. Propositional Logic: Basic Concepts**
In this chapter, we give the basic propositional concepts employed in the next

chapters of this thesis. This chapter is introduced aiming at a self contained thesis memory.

**Chapter 3. The antecedents: The Horn-CNF SAT problem**
In this chapter, we set out some basic definitions and the terminology that will be used in the thesis. In the first section we describe syntax, semantic, inference rules and a strictly linear algorithm for solving the classical two-valued Horn-CNF SAT problem. In the second section we do the same but for the many-valued Horn-CNF-SAT problem and finally in the third section we deal with the case of the SAT problem for Regular Horn-CNF propositional formulas.

**Chapter 4. The propositional Horn-NNF SAT problem**
In this chapter, we prove that the Horn-NNF-SAT problem of the two-valued propositional logic can be solved in strictly linear time. In order to proof this result, we first define syntax, semantic and inference rules for the logical system of the Horn-NNF formulas we introduce here. Next, we develop a strictly linear algorithm to solve the SAT problem in this class of formulas. In order to prove the correctness of the algorithm we apply a methodology which proves progressively the logical properties of the final and complicated linear algorithm. This methodology is firstly applied to a sub-case of the Horn-NNF-SAT problem which we call the Simple-Horn-NNF-SAT problem and afterwards it is extended to the general Horn-NNF-SAT problem.

**Chapter 5. The Regular Horn-NNF SAT problem**
In this chapter, we define two Many-valued Non-clausal Horn-like SAT problems: the Regular Simple-Horn-NNF SAT problem and the Regular Horn-NNF SAT problem. These problems are solved efficiently in $O(n \cdot log(n))$ and $O(n^2)$ time respectively. Thus, we have generalized some existing results about many-valued clausal tractability to the more general many-valued non-clausal framework. The non-clausal formulas considered here could be of significant interest in applications because they present a Horn-like structure. An important advantage of the proposed method is that it does not need to transform the original formula. Indeed, it processes the original formula preserving in this way all its logical properties contrarily to what happens when the formula is transformed to clausal forms by introducing artificial literals.

**Chapter 6. Conclusions**
In this chapter, we summarize the previous chapters and conclude on the contributions provided in this thesis.