On the Complexity of Branching-Time Logics*

Volker Weber**
Fakultät für Informatik, Technische Universität Dortmund
44221 Dortmund, Germany
volker.weber@tu-dortmund.de

Abstract. We classify the complexity of the satisfiability problem for extensions of CTL and UB. The extensions we consider are Boolean combinations of path formulas, fairness properties, past modalities, and forgettable past. Our main result shows that satisfiability for CTL with all these extensions is still in \(2\text{EXPTIME}\), which strongly contrasts with the nonelementary complexity of CTL* with forgettable past. We give a complete classification of combinations of these extensions, yielding a dichotomy between extensions with \(2\text{EXPTIME}\)-complete and those with \(\text{EXPTIME}\)-complete complexity. In particular, we show that satisfiability for the extension of UB with forgettable past is complete for \(2\text{EXPTIME}\), contradicting a claim for a stronger logic in the literature. The upper bounds are established with the help of a new kind of pebble automata.

Keywords: branching-time logic, CTL, complexity of satisfiability, pebble automata, alternating tree automata, forgettable past.

1 Introduction

Branching-time logics like CTL are an important framework for the specification and verification of concurrent and reactive systems [6,13,1]. Their history reaches almost thirty years back, when Lamport discussed the differences between linear-time and branching-time semantics of temporal logics in 1980 [19]. The first branching-time logic, called UB, was proposed the year after by Ben-Ari, Pnueli, and Manna, introducing the concept of existential and universal path quantification [2]. By extending UB with the “until” modality, Clarke and Emerson obtained the computational tree logic CTL [5], the up to date predominant branching-time logic.

Since then, many extensions of these logics have been considered. Some of these extensions aimed at more expressive power, others were introduced with the intention to make specification easier. In this paper, we consider four of these extensions that have been discussed at length in the literature, namely Boolean combinations of path formulas, fairness, past modalities, and forgettable past.

Combining these extensions, we obtain a wealth of branching-time logics. Many of the logics have been studied for their expressive power, the complexity of

* Supported by DFG grant SCHW 678/4-1.
** Deceased.
their satisfiability and model checking problems, and for optimal model checking algorithms. Nevertheless, for most of these logics the picture is still incomplete.

In this work, we complete the picture for the complexity of the satisfiability problem. Concretely, we completely classify the complexity of satisfiability for all branching-time logics obtained from UB and CTL by any combination of the extensions listed above.

Let us take a look at those parts of the picture that are already there. The classical results in the area are the proofs of EXPTIME-completeness for satisfiability of UB [2] and CTL [8]. In the following paragraphs, we review known results for the extensions we consider.

**Boolean Combinations of Path Formulas.** Both, UB and CTL, require that every temporal operator is immediately preceded by a path quantifier. Emerson and Halpern were the first to study a logic that also allows Boolean combinations of temporal operators, i.e., of path formulas, as in $E(Fp \land \neg Fq)$ [8]. They called these logics UB$^+$ and CTL$^+$ and obtained the following hierarchy on their expressive power: $UB \prec UB^+ \prec CTL \equiv CTL^+$. Concerning complexity, CTL$^+$ has been shown to be complete for 2EXPTIME by Johannsen and Lange [15]. The precise complexity of UB$^+$ is unknown.

**Fairness.** CTL cannot express fairness properties, e.g., that there exists a path on which a proposition $p$ holds infinitely often. Therefore, Emerson et al. introduced ECTL by extending CTL with a new temporal operator $F^\infty$, such that $EF^\infty p$ expresses the property above. The logic combining ECTL with the extension discussed before, ECTL$^+$, roughly corresponds to the logic CTF of [7].

The logic CTL$^*$ of Emerson and Halpern extends ECTL$^+$ with nesting of temporal operators as in $EG(p \lor Xp)$ [9]. Satisfiability for CTL$^*$, and therefore for ECTL$^+$, is 2EXPTIME-complete [26,11].

**Past Modalities.** While being common in linguistics and philosophy, past modalities are mostly viewed only as means to make specification more intuitive in computer science. For a discussion of this issue and of the possible different semantics of past modalities, we refer to [16,21]. We adopt the view of a linear, finite, and cumulative past, which is reflected in our definition of semantics of branching-time logics based on computation trees.

We use PCTL to refer to the extension of CTL with the past counterparts of the CTL temporal operators, and likewise for other logics. While PCTL is strictly more expressive than CTL [16], this is not the case for PCTL$^*$ and CTL$^*$ [14,20]. In both cases, past modalities do not increase the complexity: PCTL is EXPTIME-complete [16] and PCTL$^*$ has recently been shown to be 2EXPTIME-complete by Bozzelli [3].

**Forgettable Past.** Once past modalities are available, restricting their scope is a natural way to facilitate their use in specification. To this end, Laroussinie and Schnoebelen introduced a new operator $N$ for “from now on” to forget about

---

1 The complexity of a logic always refers to the complexity of its satisfiability problem.