Graded Computation Tree Logic ⋆

Alessandro Bianco, Fabio Mogavero, and Aniello Murano

Universitá degli Studi di Napoli "Federico II", via Cinthia, I-80126, Napoli, Italy
{alessandrobianco, mogavero, murano}@na.infn.it

1 Extended Abstract

Temporal logic is a suitable framework for reasoning about the correctness of concurrent programs. Depending on the view of the underlying nature of time, two types of temporal logics are mainly considered. In *linear-time temporal logics*, such as LTL [Pnu77], time is treated as if each moment in time has a unique possible future. Conversely, in *branching-time temporal logics*, such as CTL [CE81] and CTL* [EH86], each moment in time may split into various possible futures and *existential* and *universal quantifiers* are used to express properties along one or all the possible futures. In modal logics, such as $\mathcal{ALC}$ and μ-CALCULUS, these kinds of quantifiers have been generalized by means of *graded (worlds) modalities* [KSV02, BLMV08], which allow to express properties such as "there exist at least n accessible worlds satisfying a certain formula" or "all but n accessible worlds satisfy a certain formula". This generalization has been proved to be very powerful as it allows to express system specifications in a very succinct way. In some cases, the extension makes the logic much more complex. An example is the guarded fragment of the first order logic, which becomes undecidable when extended with a very weak form of counting quantifiers [Grä99]. In some other cases, one can extend a logic with very strong forms of "counting quantifiers" without increasing the computational complexity of the obtained logic. For example, this is the case for Gμ-CALCULUS [KSV02, BLMV08], for which the decidability problem is EXPTIME-COMPLETE.

Despite its high expressive power, the μ-CALCULUS is considered in some sense a low-level logic, making it an "unfriendly" logic for users, whereas simpler logics, such as CTL, can naturally express complex properties of computation trees. Therefore, an interesting and natural question that arises is how the extension of CTL with graded modalities can affect its expressiveness and decidability. There is a technical challenge involved in such an extension, which makes this task non-trivial: in the μ-CALCULUS, and other modal logics studied in the graded context so far, the existential and universal quantifiers range over the set of successors, thus it is easy to count the domain and its elements. In CTL, on the other hand, the underlying objects are both states and paths. Thus, the concept of graded must relapse on both of them. We solve this problem by introducing *graded path modalities* that extend to *minimal* and *conservative* paths the generalization induced to successor worlds by classical graded modalities, i.e., they allow to express properties such as "there are at least n minimal and conservative paths satisfying a formula", for suitable and well-formed concepts of minimality and conservativeness among paths. We call the logic CTL extended with graded path modalities GCTL, for short. The minimality property allows to consider only the part of a system behavior that is effectively responsible for the satisfiability of a given GCTL formula. Moreover, the minimality allows the graded path modalities to subsume the graded world modalities introduced for the μ-CALCULUS.

The introduced framework of graded path modalities turns out to be very efficient in terms of expressiveness and complexity. Indeed, we prove that GCTL is more expressive than CTL, it retains the tree and the finite model properties, and its satisfiability problem is solvable in EXPTIME, therefore not harder than that for CTL [EH85]. This, along with the fact that GCTL is exponentially more succinct than Gμ-CALCULUS, makes GCTL even more appealing.

The upper bound for the satisfiability complexity result is obtained by exploiting an automata-theoretic approach [KVW00]. To develop a decision procedure for a logic with the tree model property, one first develops an appropriate notion of tree automata and studies their emptiness problem. Then, the

⋆ This work is based on the papers [BMM09] and [BMM10], which respectively appeared in LICS 2009 and CSL 2010.

satisfiability problem for the logic is reduced to the emptiness problem of the automata. In CTL, for example, we have that every satisfiable formula admits a tree model with a polynomial branching degree and its satisfiability can be exponentially reduced to the (polynomial) emptiness of a suitable *nondeterministic Büchi tree automata* (NBT) [KVW00]. The above reasoning is much more complicate when applied to GCTL, especially when the grading numbers in the formula are coded in binary. Technically, this is due to the fact that, form one side, satisfiable GCTL formulas may require tree models with a branching degree exponential in the highest degree of the formula, and from the other side, that we have to ensure that the NBT corresponding to the formula has at most an exponential number of state w.r.t the binary coding of the highest degree of the formula.

To take care of the first point, we have develop a sharp binary encoding of each tree model. In practice, we encode each level of size n of the tree model in a binary tree of height n, by "rotating" it. To address the second point, we exploit a careful construction of the required automaton accepting all models of the formula. Basically, we use alternating tree automata enriched with *satellites* (ATAS) as an extension of that introduced in [KV06], along with the Büchi acceptance condition (ABTS). The satellite is a nondeterministic tree automaton and is used to ensure that the tree model satisfies some structural properties along its paths and it is kept apart from the main automaton. This separation, as it has been proved in [KV06], allows to solve the emptiness problem for Büchi automata in a time exponential in the number of states of the main automaton and polynomial in the number of states of the satellite. Then, we obtain the desired complexity by forcing the satellite to take care of the graded modalities and by noting that the main automaton is polynomial in the size of the formula.

Related work Graded modalities along with CTL have been also studied in [FNP08], but under a different semantics. There, the authors consider overlapping paths (as we do) as well as disjoint paths, but they do not consider the concepts of minimality and conservativeness, which we deeply use in our logics. In [FNP08] the model checking problem for non-minimal and non-conservative unary GCTL has been investigated. In particular, by opportunely extending the classical algorithm for CTL, they show that, in the case of overlapping paths, the model checking problem is PTIME-COMPLETE (thus not harder than CTL), while in the case of disjoint paths, it is in PSPACE and both NPTIME-HARD and CONPTIME-HARD. Regarding the comparison between GCTL and graded CTL with overlapping paths studied in [FNP08], it can be shown that they are equivalent by using an exponential reduction in both ways, whereas we do not know whether any of the two blow-up can be avoid. However, it is important to note that our technique can be also adapted to obtain an EXPTIME satisfiability procedure for the binary graded CTL under the semantics proposed in [FNP08].

References

[BLMV08] P.A. Bonatti, C. Lutz, A. Murano, and M.Y. Vardi. The Complexity of Enriched μ-Calculi. *LMCS*, 4(3:11):1–27, 2008.

[BMM09] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic. In *LICS'09*, pages 342–351, 2009.

[BMM10] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic with Binary Coding. In *CSL'09*, 2010.

[CE81] E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer-Verlag, 1981.

[EH85] E.A. Emerson and J.Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *JCSS*, 30(1):1–24, 1985.

[EH86] E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.

[FNP08] A. Ferrante, M. Napoli, and M. Parente. CTL Model-Checking with Graded Quantifiers. In *ATVA'08*, LNCS 5311, pages 18–32. Springer-Verlag, 2008.

[Grä99] Erich Grädel. On The Restraining Power of Guards. *JSL*, 64(4):1719–1742, 1999.

[KSV02] O. Kupferman, U. Sattler, and M.Y. Vardi. The Complexity of the Graded μ-Calculus. In *CADE'02*, LNCS 2392, pages 423–437. Springer-Verlag, 2002.

[KV06] O. Kupferman and M.Y. Vardi. Memoryful Branching-Time Logic. In *LICS'06*, pages 265–274. IEEE Computer S., 2006.

[KVW00] O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.

[Pnu77] A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57, 1977.