Efficient Normalization of Linear Temporal Logic

JAVIER ESPARZA, Technical University of Munich, Garching bei München, Germany
RUBÉN RUBIO, Universidad Complutense de Madrid, Madrid, Spain
SALOMON SICKERT, The Hebrew University, Jerusalem, Israel

In the mid 1980s, Lichtenstein, Pnueli, and Zuck proved a classical theorem stating that every formula of Past LTL (the extension of Linear Temporal Logic (LTL) with past operators) is equivalent to a formula of the form $\bigwedge_{i=1}^{n} \mathbf{GF}\varphi_i \vee \mathbf{FG}\psi_i$, where φ_i and ψ_i contain only past operators. Some years later, Chang, Manna, and Pnueli built on this result to derive a similar normal form for LTL. Both normalization procedures have a non-elementary worst-case blow-up, and follow an involved path from formulas to counter-free automata to star-free regular expressions and back to formulas. We improve on both points. We present direct and purely syntactic normalization procedures for LTL, yielding a normal form very similar to the one by Chang, Manna, and Pnueli, that exhibit only a single exponential blow-up. As an application, we derive a simple algorithm to translate LTL into deterministic Rabin automata. The algorithm normalizes the formula, translates it into a special very weak alternating automaton, and applies a simple determinization procedure, valid only for these special automata.

CCS Concepts: • **Theory of computation** → **Modal and temporal logics**; **Automata over infinite objects;**

Additional Key Words and Phrases: Linear temporal logic, normal form, weak alternating automata, deterministic automata

ACM Reference Format:
Javier Esparza, Rubén Rubio, and Salomon Sickert. 2024. Efficient Normalization of Linear Temporal Logic. *J. ACM* 71, 2, Article 16 (April 2024), 42 pages. https://doi.org/10.1145/3651152

1 INTRODUCTION

In the late 1970s, Amir Pnueli introduced **Linear Temporal Logic (LTL)** into computer science as a framework for specifying and verifying concurrent programs [40, 41], a contribution that earned him the 1996 Turing Award. During the 1980s and the early 1990s, Pnueli et al. proceeded to study the properties expressible in LTL. In 1985, Lichtenstein et al. [24] introduced a classification of LTL properties, later described in detail by Manna and Pnueli [29, 30] in two famous monographs,

This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) under projects 183790222, 317422601, and 436811179; by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement 787367 (PaVeS); by the Agencia Estatal de Investigación (AEI) under project PID2019-108528RB-C22; and by the Spanish Ministry of Universities under grants FPU17/02319 and EST21/00536.

Authors' addresses: J. Esparza, Technical University of Munich, Fakultät für Informatik, Boltzmannstrasse 3, Garching bei München, 85748, Germany; e-mail: esparza@in.tum.de; R. Rubio, Universidad Complutense de Madrid, Facultad de Informática, Calle Profesor José García Santesmases 9, Madrid, 28040, Spain; e-mail: rubenrub@ucm.es; S. Sickert, The Hebrew University, School of Computer Science and Engineering, Rothberg Family Buildings, The Edmond J. Safra Campus, Jerusalem, 91904, Israel; e-mail: s.sickert@tum.de.

ACM 0004-5411/2024/04-ART16
https://doi.org/10.1145/3651152

which also gave it its current name, the *safety-progress* hierarchy (see also the work of Piterman and Pnueli [39] for a brief account). The safety-progress hierarchy consists of a *safety* class of formulas and five *progress* classes. The classes are defined semantically in terms of their models, and the largest class, called the *reactivity* class in the work of Manna and Pnueli [29, 30], contains all properties expressible in LTL. Manna and Pnueli provide syntactic characterizations of each class. In particular, they state a fundamental theorem showing that every reactivity property is expressible as a conjunction of formulas of the form $\mathbf{GF}\varphi \vee \mathbf{FG}\psi$, where $\mathbf{F}\chi$ and $\mathbf{G}\chi$ mean that χ holds at some and at every point in the future, respectively, and φ, ψ only contain past operators. Manna and Pnueli call this the *normal form for Past LTL*.

Technically, the preceding works consider Past LTL, an extension of LTL with past operators. In 1992, Chang et al. [8] presented a different and very elegant characterization of the safety-progress hierarchy in terms of standard LTL, the logic containing only the future operators **X** (next), **U** (until), and **W** (weak until). They showed that every reactivity formula is equivalent to an LTL formula in negation normal form such that every path through the syntax tree contains at most one alternation of **U** and **W**. We call this fundamental result the *Normalization Theorem*. In the notation of other work [6, 38, 45], which mimics the definition of the Σ_i, Π_i, and Δ_i classes of the arithmetical and polynomial hierarchies, they proved that every LTL formula is equivalent to a Δ_2-formula.

While these normal forms for LTL have had large conceptual impact in model checking, automatic synthesis, and deductive verification (see, e.g., the work of Piterman and Pnueli [39] for a survey), the normalization *procedures* for LTL have had none. The reason is that many of the known procedures are not direct, meaning that they require to translate formulas into automata and back, all have high complexity, and their correctness proofs are involved. Let us elaborate on this:

- As mentioned previously, the normal form for Past LTL is first stated by Lichtenstein et al. [24, p. 208]. It is a consequence of Theorem 2, whose proof is not given in the paper but it is only said that the proof is based on many previous results, including results from five different papers [3, 9, 17, 32, 46].
- In three chapters of her Ph.D. thesis, Zuck [49] gives a detailed description of the normalization procedure of Lichtenstein et al. [24]. First, Zuck translates the initial Past LTL formula into a counter-free automaton, then applies the Krohn-Rhodes cascade decomposition of finite automata and other results to translate the automaton into a star-free regular expression, and finally translates this expression into a reactivity formula with a non-elementary blow-up.
- The normal form for LTL is stated in Section 4.2 of the monograph of Manna and Pnueli [31, p. 296], and in their PODC 1990 survey paper [29, p. 399]. The proof is said to be out of scope, and no normalization procedure is presented.
- In the work of Chang et al. [8], the Normalization Theorem for LTL is stated. They give a rough sketch of the normalization procedure and do not give a proof. The normalization procedure uses the translation of other works [24, 49] from star-free regular expressions to Past LTL as a subroutine, and so it is at least as complex as the one of other works [24, 49].
- A normalization procedure for LTL of elementary complexity can be obtained by combining work by Maler and Pnueli [26, 28] on the Krohn-Rhodes decomposition with a recent result by Boker et al. [2] on translating automata back into LTL (when possible). The procedure has three steps. First, the formula is translated into a deterministic ω-regular automaton using, for example, the double-exponential construction of Safra [43]. Second,

this automaton is translated into an equivalent deterministic and counter-free automaton, using the single-exponential construction of Maler and Pnueli [27]. Finally, this automaton is translated into a Δ_2-formula using the triple-exponential construction of Boker et al. [2]. This combination of constructions yields an elementary normalization procedure, which moreover produces an equivalent formula at the smallest possible level of the safety-progress hierarchy. However, the procedure is indirect and has high complexity. In particular, while future work may improve the blow-up of the last two steps, the double-exponential bound on the translation of LTL into deterministic ω-automata is tight. Therefore, any procedure that constructs a deterministic ω-automaton as intermediate step must have at least double-exponential complexity.

— Reynolds [42] and Guelev [18] give direct proofs of the normalization theorem for Past LTL that do not require to translate formulas into automata. Both proofs rely on different versions of Gabbay's famous separation theorem, stating that every formula of Past LTL is equivalent to a Boolean combination of past and future formulas [15, 16]. Gabbay's theorem is proved by means of equivalence preserving syntactic transformations, and so are the proofs of Guelev [18] and Reynolds [42]. However, the only known upper bound on the blow-up in the size of the formula produced by Gabbay's separation procedure is non-elementary. Oliveira and Rasga [37] give a double-exponential separation algorithm for a fragment of Past LTL that restricts the nesting of the since and until operators.

It is remarkable that, despite the prominence of the safety-bounded hierarchy in the work of Manna and Pnueli, the complexity of normalization procedures has not been studied further, even though no lower bound for the blow-up it involves is known. In particular, and contrary to the case of propositional and first-order logic, where efficient normalization algorithms for conjunctive and clausal normal form are essential part of SAT or first-order theorem provers, normalization has not been used in LTL to obtain more efficient algorithms for satisfiability, model-checking, or synthesis tasks. This article contains three main results that, in our opinion, completely change this situation:

(1) A simple proof of the Normalization Theorem for LTL. The proof is direct (it does not require any knowledge of automata or regular expressions), gives a clear intuitive explanation of why normalization is possible, and yields a closed form for the normalized formula with *single-exponential* blow-up.
(2) An efficient normalization algorithm. The normalization procedure given in (1) has exponential *best-case* complexity, and is not goal oriented, in the sense that it does not only concentrate on those parts of the formula that do not belong to Δ_2. We provide a normalization algorithm consisting of six rewrite rules that solves these problems. In particular, the rewrite rules can be applied locally to "offending subformulas."
(3) A novel translation of LTL into deterministic Rabin automata (DRW) exploits the results of (1) and (2). The translation normalizes the formula and then transforms it into an alternating Rabin automaton with at most one alternation between accepting and non-accepting states (A1W1). This automaton is determinized by means of a novel, dedicated algorithm for A1W1, with better properties than the construction Safra [43]. In particular, the states of the DRW are pairs of sets of states of the A1W1, instead of trees of sets of states, as would be the case with Safra's construction. This simpler state structure leads to smaller DRW.

At the heart of our first result is a novel technique, interesting on its own, called *contextual normalization*. Loosely speaking, to normalize a formula φ interpreted on infinite words over some given alphabet Σ, the technique constructs a finite cover of the set Σ^ω. This is achieved by carefully selecting a set of formulas B, called a *basis*. The cover contains one set of words $L_{C|B}$

for every $C \subseteq B$, defined as the set of the words satisfying all formulas of C and none of $B \setminus C$. For every $C \subseteq B$, we find a formula $\varphi\langle C \mid B\rangle$ of Δ_2 equivalent to φ over all words of $L_{C|B}$ (i.e., every word of $L_{C|B}$ satisfies either both formulas or none). Intuitively, $\varphi\langle C \mid B\rangle$ is equivalent to φ in the context of C. Then we "patch together" these formulas to obtain a formula equivalent to φ on all words.

The second result shows that LTL formulas can also be normalized by means of a rewrite system, just as one brings a Boolean formula in CNF; the only difference is the need for *contextual* rewrite rules, specifying that a subformula of a formula can only be rewritten if the formula has a certain form.

The third result shows that, on top of the central role it plays in the work of Manna and Pnueli, normalization can lead to novel algorithms for questions in the theory and applications of LTL that continue to be investigated today. In particular, efficient translations from LTL to automata exhibiting different degrees of non-determinism are being intensely studied, due to their applications to controller synthesis and probabilistic model checking, among others (see, e.g., [4, 5, 11, 12, 20, 21]). The translation of (3) based on normalization has already become part of the Owl library for ω-automata [22], and it is used in the Strix synthesis tool [33].

The article is structured as follows. Section 2 introduces the syntax and semantics of LTL. Section 3 introduces Manna and Pnueli's safety-progress hierarchy—following the notation of Cerná and Pelánek [6]—and recalls the Normalization Theorem presented elsewhere [8, 24, 29]. Section 4 presents our novel proof of the theorem based on contextual equivalence, and in particular Theorem 4.22, our first normalization procedure. Section 5 describes a normalizing rewrite system for LTL consisting of the rules presented in Tables 1 and 2, respectively. Section 6 introduces a translation from LTL to deterministic Rabin automata based on normalization, summarized in Theorem 6.16. Section 7 shows a tight correspondence between the classes of the safety-progress hierarchy and weak alternating automata. Section 8 contains some conclusions.

Remark. This article is a revised and extended version of previous work by the authors [13, 45]. Preceding results (1) and (3) were first presented in our work from 2020 [45] and result (2) in our work from 2022 [13].

2 PRELIMINARIES

Let Σ be a finite alphabet. A *word* w over Σ is an infinite sequence of letters $a_0 a_1 a_2 \dots$ with $a_i \in \Sigma$ for all $i \geq 0$. A *finite word* is a finite sequence of letters. The set of all words (finite words) is denoted Σ^ω (Σ^*). We let $w[i]$ (starting at $i = 0$) denote the i-th letter of a word w. The finite infix $w[i]w[i+1]\dots w[j-1]$ is abbreviated with w_{ij} and the infinite suffix $w[i]w[i+1]\dots$ with w_i. We denote the infinite repetition of a finite word $a_0 \dots a_n$ by $(a_0 \dots a_n)^\omega = a_0 \dots a_n a_0 \dots a_n a_0 \dots$. A set of (finite or infinite) words is called a *language*.

2.1 Syntax and Semantics of LTL

Formulas of LTL over a finite set Ap of atomic propositions are constructed by the following syntax:

$$\varphi ::= \mathbf{tt} \mid \mathbf{ff} \mid a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \varphi\,\mathbf{W}\,\varphi \mid \varphi\,\mathbf{R}\,\varphi \mid \varphi\,\mathbf{M}\,\varphi, \tag{1}$$

where $a \in Ap$ is an atomic proposition and $\mathbf{X}$, $\mathbf{U}$, $\mathbf{W}$, $\mathbf{R}$, and $\mathbf{M}$ are the next, (strong) until, weak until, (weak) release, and strong release operators, respectively. Further, we use the standard abbreviations $\mathbf{F}\varphi := \mathbf{tt}\ \mathbf{U}\ \varphi$ (eventually) and $\mathbf{G}\varphi := \varphi\ \mathbf{W}\ \mathbf{ff}$ (always). The *size* of a formula is the number of nodes of its syntax tree.

Formulas are interpreted on words over the alphabet $\Sigma := 2^{Ap}$. Let w be such a word, and let φ be a formula. The satisfaction relation $w \models \varphi$ is inductively defined as the smallest relation satisfying:

$$\begin{array}{lll@{\qquad}lll}
w \models \mathbf{tt} & & \text{for every } w & w \models \mathbf{X}\varphi & \text{iff} & w_1 \models \varphi \\
w \not\models \mathbf{ff} & & \text{for every } w & w \models \varphi \,\mathbf{U}\, \psi & \text{iff} & \exists k.\, w_k \models \psi \text{ and } \forall j < k.\, w_j \models \varphi \\
w \models a & \text{iff} & a \in w[0] & w \models \varphi \,\mathbf{M}\, \psi & \text{iff} & \exists k.\, w_k \models \varphi \text{ and } \forall j \leq k.\, w_j \models \psi \\
w \models \neg\varphi & \text{iff} & w \not\models \varphi & w \models \varphi \,\mathbf{R}\, \psi & \text{iff} & \forall k.\, w_k \models \psi \text{ or } w \models \varphi \,\mathbf{M}\, \psi \\
w \models \varphi \wedge \psi & \text{iff} & w \models \varphi \text{ and } w \models \psi & w \models \varphi \,\mathbf{W}\, \psi & \text{iff} & \forall k.\, w_k \models \varphi \text{ or } w \models \varphi \,\mathbf{U}\, \psi. \\
w \models \varphi \vee \psi & \text{iff} & w \models \varphi \text{ or } w \models \psi & & &
\end{array}$$

We let $\mathcal{L}(\varphi) := \{w \in \Sigma^\omega : w \models \varphi\}$ denote the language of φ. Two formulas φ and ψ are *equivalent*, denoted $\varphi \equiv \psi$, if $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$. We overload the definition of $\models$ and write $\varphi \models \psi$ as a shorthand for $\mathcal{L}(\varphi) \subseteq \mathcal{L}(\psi)$.

Monotonic and Dual Operators. An operator $\mathbf{P}$ of arity $k \geq 0$ is *monotonic* if $(\varphi_1 \models \psi_1) \wedge \cdots \wedge (\varphi_k \models \psi_k)$ implies $\mathbf{P}(\varphi_1, \ldots, \varphi_k) \models \mathbf{P}(\psi_1, \ldots, \psi_k)$. It is easy to see that all operators of the syntax, with the exception of negation, are monotonic. Two operators $\mathbf{P}$ and $\mathbf{Q}$ of arity $k \geq 0$ are *dual* if $\mathbf{P}(\varphi_1, \ldots, \varphi_k) \equiv \neg\mathbf{Q}(\neg\varphi_1, \ldots, \neg\varphi_k)$ and $\mathbf{Q}(\varphi_1, \ldots, \varphi_k) \equiv \neg\mathbf{P}(\neg\varphi_1, \ldots, \neg\varphi_k)$ for all formulas $\varphi_1, \ldots, \varphi_k$. It is easy to see that $\mathbf{tt}$ and $\mathbf{ff}$, $\vee$ and $\wedge$, $\mathbf{U}$ and $\mathbf{R}$, and $\mathbf{W}$ and $\mathbf{M}$ are dual, and $\mathbf{X}$ is self dual.

Negation Normal Form. A formula is in *negation normal form* if negations appear only in front of atomic propositions. In other words, the syntax of formulas for negation normal form is obtained by substituting $\neg\varphi$ for $\neg a$ in (1).

The following result is folklore.

Proposition 2.1. *Every formula of size n has an equivalent formula in negation normal form of size $O(n)$. Further, for every formula φ in negation normal form of size n there exists a formula $\overline{\varphi}$ in negation normal form of size n such that $\neg\varphi \equiv \overline{\varphi}$.*

Proof. Formulas are put in negation normal form by "pushing negations" across all operators using the duality relations (e.g., $\neg(\varphi_1 \,\mathbf{U}\, \varphi_2)$ is replaced by $\neg\varphi_1 \,\mathbf{R}\, \neg\varphi_2$). These transformations only increase the size of the formula by one unit, and so the formula in negation normal form has size $O(n)$. The formula $\overline{\varphi}$ is defined inductively using the duality relations (e.g., $\overline{\varphi_1 \,\mathbf{U}\, \varphi_2} := \overline{\varphi_1} \,\mathbf{R}\, \overline{\varphi_2}$, $\overline{\mathbf{X}\varphi} := \mathbf{X}\overline{\varphi}$); these transformations preserve the size of the formula. □

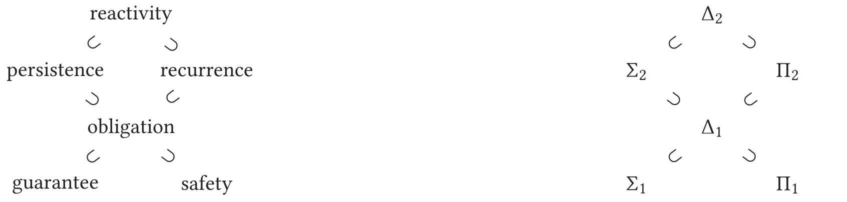
> *Convention*: In the rest of the article we assume, without explicit mention, that formulas are in negation normal form. Abusing language, we call $\overline{\varphi}$ the negation of φ.

3 THE SAFETY-PROGRESS HIERARCHY

We recall the definition of the safety-progress hierarchy, the hierarchy of temporal properties studied by Manna and Pnueli [29]. We follow the formulation of Černá and Pelánek [6]. The definition of the hierarchy formalizes the intuition that, for example, a safety property is violated by an execution iff one of its finite prefixes is "bad" or, equivalently, satisfied by an execution iff all of its finite prefixes belong to a language of good prefixes. In the ensuing sections, we describe structures that have a direct correspondence to this hierarchy, and in this sense the hierarchy provides a map to navigate the results of this work.

Definition 3.1 ([6, 29]). Let $P \subseteq \Sigma^\omega$ be a property over Σ:

- P is a safety property if there exists a language of finite words $L \subseteq \Sigma^*$ such that $w \in P$ iff all finite prefixes of w belong to L.



(a) Safety-progress hierarchy [30]

(b) Syntactic-future hierarchy

Fig. 1. Both hierarchies, side by side, indicating the correspondence of Theorem 3.4 for properties specifiable in LTL.

- P is a guarantee property if there exists a language of finite words $L \subseteq \Sigma^*$ such that $w \in P$ iff there exists a finite prefix of w which belongs to L.
- P is an obligation property if it can be expressed as a positive Boolean combination of safety and guarantee properties.
- P is a recurrence property if there exists a language of finite words $L \subseteq \Sigma^*$ such that $w \in P$ iff infinitely many prefixes of w belong to L.
- P is a persistence property if there exists a language of finite words $L \subseteq \Sigma^*$ such that $w \in P$ iff all but finitely many prefixes of w belong to L.
- P is a reactivity property if P can be expressed as a positive Boolean combination of recurrence and persistence properties.

The inclusions between these classes are shown in Figure 1(a). Chang et al. [8] give a syntactic characterization of the classes in terms of the following fragments of LTL.

Definition 3.2 (Adapted from Cerná and Pelánek [6]). We define the following classes of LTL formulas:

- The class $\Sigma_0 = \Pi_0 = \Delta_0$ is the least set of formulas containing $\mathbf{tt}$, $\mathbf{ff}$, all atomic propositions and their negations, and is closed under the application of conjunction and disjunction.
- The class Σ_{i+1} is the least set of formulas containing Π_i that is closed under the application of conjunction, disjunction, and the $\mathbf{X}$, $\mathbf{U}$, and $\mathbf{M}$ operators.
- The class Π_{i+1} is the least set of formulas containing Σ_i that is closed under the application of conjunction, disjunction, and the $\mathbf{X}$, $\mathbf{R}$, and $\mathbf{W}$ operators.
- The class Δ_{i+1} is the least set of formulas containing Σ_{i+1} and Π_{i+1} that is closed under the application of conjunction and disjunction.

Observe the behavior of the classes under negation. Given a set of formulas F, let $\overline{F} = \{\overline{\varphi} \mid \varphi \in F\}$. By the definition of $\overline{\phi}$, we have the following proposition.

Proposition 3.3. *For every $i \geq 0$, we have $\overline{\Sigma}_i = \Pi_i$, $\overline{\Pi}_i = \Sigma_i$, and $\overline{\Delta}_i = \Delta_i$.*

In particular, Proposition 3.3 shows that a formula φ is equivalent to a formula of Δ_2 iff $\overline{\varphi}$ is.

The following result, a corollary of the proof of Chang et al. [8, Theorem 8], shows that the safety-progress hierarchy and the syntactic hierarchy of Definition 3.2 coincide.

Theorem 3.4 (Adapted from Cerná and Pelánek [6]). *A property that is specifiable in LTL is a guarantee (safety, obligation, persistence, recurrence, reactivity, respectively) property iff it is specifiable by a formula from the class $\Sigma_1(\Pi_1, \Delta_1, \Sigma_2, \Pi_2, \Delta_2$, respectively).*

Together with the result of Lichtenstein et al. [24], stating that every formula of LTL is equivalent to a reactivity formula, Chang et al. obtain the following theorem.

Theorem 3.5 (Normalization Theorem [8, 24, 29]). *Every LTL formula is equivalent to a formula of* Δ_2.

4 A SIMPLE PROOF OF THE NORMALIZATION THEOREM

We present a simple proof of the Normalization Theorem. Section 4.1 introduces our proof technique, the *Contextual Equivalence Lemma*. The lemma shows that every formula φ is normalizable if certain formulas related to φ exist. Sections 4.3 and 4.4 prove the existence of those formulas. Finally, Section 4.5 instantiates the Contextual Equivalence Lemma with the formulas of Sections 4.3 and 4.4, which yield the theorem.

4.1 Contextual Equivalence

Consider the formula $\varphi = \mathbf{FG}(a \mathbin{\mathbf{U}} b)$. It does not belong to Δ_2, because of the alternation **F**-**G**, followed by the alternation **G**-**U**. Let us see how to find an equivalent formula of Δ_2. We consider the formula $\mathbf{GF}b$ and argument (informally!) as follows:

- For words that do not satisfy $\mathbf{GF}b$, φ is equivalent to **ff**. Indeed, if $w \not\models \mathbf{GF}b$, then b holds for only finitely many suffixes of w, and so $a \mathbin{\mathbf{U}} b$ also holds for finitely many suffixes only. So w cannot satisfy $\mathbf{FG}(a \mathbin{\mathbf{U}} b)$.
- For words that satisfy $\mathbf{GF}b$, φ is equivalent to $\mathbf{FG}(a \mathbin{\mathbf{W}} b)$. Indeed, if a word w satisfies $\mathbf{GF}b$, then every suffix w_i satisfies $\mathbf{F}b$, and so $w_i \models a \mathbin{\mathbf{U}} b$ iff $w_i \models a \mathbin{\mathbf{W}} b$ holds for every $i \geq 0$.

We say that φ is equivalent to $\mathbf{FG}(a \mathbin{\mathbf{W}} b)$ *in the context of* $\mathbf{GF}b$, and equivalent to **ff** *in the context of* $\overline{\mathbf{GF}b}$. We get $\mathbf{FG}(a \mathbin{\mathbf{U}} b) \equiv (\mathbf{GF}b \wedge \mathbf{FG}(a \mathbin{\mathbf{W}} b)) \vee (\overline{\mathbf{GF}b} \wedge \mathbf{ff}) \equiv (\mathbf{GF}b \wedge \mathbf{FG}(a \mathbin{\mathbf{W}} b))$. Due to the elimination of **U**, this is a formula of Δ_2.

We can proceed dually with the formula $\varphi = \mathbf{GF}(a \mathbin{\mathbf{W}} b)$. We consider the formula $\mathbf{FG}a$ and argument as follows:

- In the context of $\mathbf{FG}a$, φ is equivalent to **tt**. Indeed, if $w \models \mathbf{FG}a$, then there is a suffix w_i such that $a \mathbin{\mathbf{W}} b$ holds for all suffixes of w_i, and so in particular for infinitely many suffixes.
- In the context of $\overline{\mathbf{FG}a}$, φ is equivalent to $\mathbf{GF}(a \mathbin{\mathbf{U}} b)$. Recall that $a \mathbin{\mathbf{W}} b \equiv a \mathbin{\mathbf{U}} b \vee \mathbf{G}a$. If a word w does not satisfy $\mathbf{FG}a$, then no suffix satisfies $\mathbf{G}a$; so w satisfies φ iff infinitely many prefixes satisfy $a \mathbin{\mathbf{U}} b$.

Now we have $\mathbf{GF}(a \mathbin{\mathbf{W}} b) \equiv (\mathbf{FG}a \wedge \mathbf{tt}) \vee (\overline{\mathbf{FG}a} \wedge \mathbf{GF}(a \mathbin{\mathbf{U}} b)) \equiv \mathbf{FG}a \vee (\overline{\mathbf{FG}a} \wedge \mathbf{GF}(a \mathbin{\mathbf{U}} b)) \equiv \mathbf{FG}a \vee (\mathbf{GF}(\neg a) \wedge \mathbf{GF}(a \mathbin{\mathbf{U}} b)) \equiv \mathbf{FG}a \vee \mathbf{GF}(a \mathbin{\mathbf{U}} b)$. Due to the elimination of **W**, this is a formula of Δ_2.

Our normalization strategy follows this pattern. Given a formula φ, we define a *basis* of formulas B and consider all *contexts* $\langle C \mid B \rangle$, where $C \subseteq B$. Intuitively, a context $\langle C \mid B \rangle$ corresponds to a region of the set of all words containing the words satisfying all formulas of C and none of $B \setminus C$, or, equivalently, all dual formulas of the formulas in $B \setminus C$. For every context $\langle C \mid B \rangle$, we find a formula $\varphi\langle C|B\rangle$ of Δ_2 equivalent to φ over all words of the corresponding region. Then we "patch together" these formulas to obtain a formula of Δ_2 that is equivalent to φ everywhere.

We formalize this idea in two steps. First we state and prove the Weak Contextual Equivalence Lemma, which is enough to normalize simple formulas like $\mathbf{FG}(a \mathbin{\mathbf{U}} b)$ and $\mathbf{GF}(a \mathbin{\mathbf{W}} b)$. In the second step, we state and prove the Contextual Equivalence Lemma, which presents a technique to normalize arbitrary formulas.

Fix a set of atomic propositions Ap, and let $\mathcal{U}$ be the set of all words over 2^{Ap}.

Definition 4.1 (Basis and Equivalence under Context). Let B be a finite set of formulas over Ap, called a *basis*. A *context* is a partition of B into a set $C \subseteq B$ and the set $B \setminus C$. We denote a context by $\langle C|B\rangle$. The language $L_{C|B} \subseteq \mathcal{U}$ of $\langle C|B\rangle$ is the set of words that satisfy every formula of C and no formula of $B \setminus C$. Two formulas φ_1, φ_2 are *equivalent under context* $\langle C \,|\, B\rangle$ if $(w \models \varphi_1 \Leftrightarrow w \models \varphi_2)$ holds for every $w \in L_{C|B}$.

We collect some simple properties for later use.

Lemma 4.2. *We have the following:*

(1) $L_{\emptyset|\emptyset} = \mathcal{U}$. In particular, two formulas are equivalent iff they are equivalent under context $\langle \emptyset \,|\, \emptyset\rangle$.
(2) Let $B \subseteq B'$. Then $L_{C'|B'} \subseteq L_{(C' \cap B)|B}$ for every $C' \subseteq B'$.
(3) If two formulas are equivalent under context $\langle C \mid B\rangle$, then they are also equivalent under any context $\langle C' \,|\, B'\rangle$ such that $C = C' \cap B$ and $B \subseteq B'$.
(4) Let $\overline{B} := \{\overline{\psi} \mid \psi \in B\}$ be the dual *basis of B. For every $C \subseteq B$, define $C_d := \{\overline{\psi} \mid \psi \in B \setminus C\}$. Two formulas are equivalent under context $\langle C \,|\, B\rangle$ iff they are equivalent under context $\langle C_d \,|\, \overline{B}\rangle$.*

Proof. We proceed as follows:

(1) Follows immediately from the definition.
(2) Let $w \in L_{C'|B'}$. By definition, we have $w \models \psi$ for all $\psi \in C'$ and $w \not\models \psi$ for all $\psi \in B' \setminus C'$. We have $B \setminus (C' \cap B) = B \setminus C' \subseteq B' \setminus C'$. So $w \models \psi$ for all $\psi \in C' \cap B$ and $w \not\models \psi$ for all $\psi \in B \setminus (C' \cap B)$. By definition, $w \in L_{(C' \cap B)|B}$.
(3) Let φ_1, φ_2 be equivalent under $\langle C \,|\, B\rangle$. We have $w \models \varphi_1 \Leftrightarrow$ iff $w \models \varphi_2$ for every $w \in L_{C|B}$. By $B \subseteq B'$, $C = C' \cap B$, and (2), we get $L_{C'|B'} \subseteq L_{C|B}$, and so φ_1, φ_2 are equivalent under $\langle C' \,|\, B'\rangle$.
(4) It suffices to prove $L_{C_d|\overline{B}} = L_{C|B}$. This is an easy consequence of $\overline{\psi} \equiv \neg\psi$. Indeed, for any $w \in \mathcal{U}$, we have

$$w \models \psi \text{ for all } \psi \in C \quad \text{iff} \quad w \not\models \overline{\psi} \text{ for all } \psi \in C \quad \text{iff} \quad w \not\models \overline{\psi} \text{ for all } \psi \in B \setminus (B \setminus C) \quad \text{iff} \\ w \not\models \overline{\psi} \text{ for all } \psi \in \overline{B} \setminus C_d.$$

Similarly, $w \not\models \psi$ for all $\psi \in B \setminus C$ iff $w \models \overline{\psi}$ for all $\psi \in C_d$. Hence, $w \in L_{C|B}$ iff $w \in L_{C_d|\overline{B}}$.

□

4.1.1 Weak Contextual Equivalence Lemma. Intuitively, the Weak Contextual Equivalence Lemma shows how to patch together formulas $\varphi\langle C \,|\, B\rangle$ that are equivalent to φ for each context $\langle C \,|\, B\rangle$.

Lemma 4.3 (Weak Contextual Equivalence Lemma). *Let φ be a formula, and let B be a basis of formulas. Assume that for every context $\langle C \,|\, B\rangle$ there exists a formula $\varphi\langle C \,|\, B\rangle$ satisfying*

(i) φ and $\varphi\langle C \,|\, B\rangle$ are equivalent under context $\langle C \,|\, B\rangle$, and
(ii) for every $C, C' \subseteq B$, if $C \subseteq C'$, then $\varphi\langle C \,|\, B\rangle \models \varphi\langle C' \,|\, B\rangle$.

Then we have

$$\varphi \equiv \bigvee_{C \subseteq B} \left(\varphi\langle C \,|\, B\rangle \wedge \bigwedge_{\psi \in C} \psi \right).$$

Proof. Let $w \in \mathcal{U}$. Let C_w be the (possibly empty) set of formulas of the basis B satisfied by w.

Assume $w \models \varphi$. We have $w \models \bigwedge_{\psi \in C_w} \psi$ and $w \in L_{C_w|B}$ by definition of C_w, and so $w \models \varphi\langle C_w \,|\, B\rangle$ by property (i). So $w \models \varphi\langle C_w \,|\, B\rangle \wedge \bigwedge_{\psi \in C_w} \psi$, and thus $w \models \bigvee_{C \subseteq B} \left(\varphi\langle C \,|\, B\rangle \wedge \bigwedge_{\psi \in C} \psi\right)$.

Assume $w \models \bigvee_{C \subseteq B} \left(\varphi\langle C \,|\, B\rangle \wedge \bigwedge_{\psi \in C} \psi\right)$. Then there exists a context $C \subseteq B$ such that $w \models \varphi\langle C \,|\, B\rangle \wedge \bigwedge_{\psi \in C} \psi$. By the definition of C_w, we have $C \subseteq C_w$, and so $w \models \varphi\langle C_w \mid B\rangle$ by property (ii). Since, by property (i), φ and $\varphi\langle C_w \,|\, B\rangle$ are equivalent under context $\langle C_w \,|\, B\rangle$, we get $w \models \varphi$. □

Example 4.4. Consider the formula $\varphi := \mathbf{FG}(a\mathbf{U}b)$. It is easy to see that $B := \{\mathbf{GF}b\}$, $\varphi\langle\emptyset\,|\,B\rangle := \mathbf{ff}$ and $\varphi\langle B\,|\,B\rangle := \mathbf{FG}(a\;\mathbf{W}\;b)$ satisfy the conditions of the lemma. So we have

$$\varphi \equiv (\varphi\langle\emptyset\,|\,B\rangle \wedge \mathbf{tt}) \vee (\varphi\langle B\,|\,B\rangle \wedge \mathbf{GF}b) = (\mathbf{ff} \wedge \mathbf{tt}) \vee (\mathbf{FG}(a\;\mathbf{W}\;b) \wedge \mathbf{GF}b) \equiv \mathbf{FG}(a\;\mathbf{W}\;b) \wedge \mathbf{GF}b \in \Delta_2.$$

Consider now $\varphi := \mathbf{GF}(a\;\mathbf{W}\;b)$. Taking $B := \{\mathbf{FG}a\}$, $\varphi\langle\emptyset\,|\,B\rangle := \mathbf{GF}(a\;\mathbf{U}\;b)$, and $\varphi\langle B\,|\,B\rangle := \mathbf{tt}$ also satisfies the conditions, and we get

$$\varphi \equiv (\varphi\langle\emptyset\,|\,B\rangle \wedge \mathbf{tt}) \vee (\varphi\langle B\,|\,B\rangle \wedge \mathbf{FG}a) = (\mathbf{GF}(a\;\mathbf{U}\;b) \wedge \mathbf{tt}) \vee (\mathbf{tt} \wedge \mathbf{FG}a) \equiv \mathbf{GF}(a\;\mathbf{U}\;b) \vee \mathbf{FG}a \in \Delta_2.$$

The Weak Contextual Equivalence Lemma only shows how to normalize formulas under the assumption that *one already knows how to normalize the formulas of the basis* B. Indeed, if B contains formulas that are not in Δ_2, then for any $C \subseteq B$ containing such formulas, the expression $\bigwedge_{\psi \in C} \psi$ is also not in Δ_2. But how can one normalize these formulas of B? The obvious idea is to recursively apply the lemma: find a second basis B' of "simpler" formulas, reduce the problem of normalizing B to normalizing B', and iterate until a basis containing only formulas of Δ_2 is reached. The Contextual Equivalence Lemma follows this idea but also improves on it; instead of a sequence of bases, it only assumes a unique *well-founded* basis.

4.1.2 *Contextual Equivalence Lemma.* We introduce well-founded bases.

Definition 4.5. Let B be a basis and $(\prec) \in B \times B$ a well-founded order on B. We say that $(B, \prec)$ is a *well-founded basis*. Given a basis formula $\psi \in B$, we let $\psi{\downarrow}$ denote the set $\{\chi \in B \mid \chi \prec \psi\}$.

Intuitively, after applying weak contextual equivalence to φ with basis B, we may need to apply it again to ψ. But with which basis? The answer is with the (well-founded) basis $\psi{\downarrow}$. Well-foundedness guarantees that this process terminates. The Contextual Equivalence Lemma also adds to the assumptions (i) and (ii) on φ similar assumptions (iii) and (iv) on the formulas $\psi \in B$.

Lemma 4.6 (Contextual Equivalence Lemma). *Let φ be a formula, and let $(B, \prec)$ be a well-founded basis. Assume that for every $C \subseteq B$ there exists a formula $\varphi\langle C\,|\,B\rangle$ satisfying*

(i) φ and $\varphi\langle C\,|\,B\rangle$ are equivalent under context $\langle C\,|\,B\rangle$, and
(ii) for every two sets $C, C' \subseteq B$, if $C \subseteq C'$, then $\varphi\langle C\,|\,B\rangle \models \varphi\langle C'\,|\,B\rangle$.

Assume further that for every basis formula $\psi \in B$ and every $C \subseteq \psi{\downarrow}$ there exists a formula $\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle$ satisfying

(iii) ψ and $\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle$ are equivalent under context $\langle C\,|\,\psi{\downarrow}\rangle$, and
(iv) for every two sets $C, C' \subseteq \psi{\downarrow}$, if $C \subseteq C'$, then $\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle \models \psi\langle\!\langle C'\,|\,\psi{\downarrow}\rangle\!\rangle$.

Then we have

$$\varphi \equiv \bigvee_{C \subseteq B} \left(\varphi\langle C\,|\,B\rangle \wedge \bigwedge_{\psi \in C} \psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle \right).$$

Proof. Let w be a word. We prove $w \models \varphi$ iff $w \models \bigvee_{C \subseteq B} \left(\varphi\langle C \mid B\rangle \wedge \bigwedge_{\psi \in C} \psi\langle\!\langle C \mid \psi{\downarrow}\rangle\!\rangle\right)$. Let $C_w \subseteq B$ be the (possibly empty) set of formulas of B satisfied by w.

($\Rightarrow$): Assume $w \models \varphi$. Observe that $w \in L_{C_w|B}$ by definition of C_w. We prove that $w \models \varphi\langle C_w \mid B\rangle$ and $w \models \psi\langle\!\langle C_w\,|\,\psi{\downarrow}\rangle\!\rangle$ for every $\psi \in C_w$:

- $w \models \varphi\langle C_w \mid B\rangle$. Since $w \in L_{C_w|B}$ and $w \models \varphi$ by hypothesis, we have $w \models \varphi\langle C_w \mid B\rangle$ by property (i).
- $w \models \psi\langle\!\langle C_w \mid \psi\,{\downarrow}\rangle\!\rangle$ for every $\psi \in C_w$. Let $\psi \in C_w$. We have $w \in L_{C_w|B} \subseteq L_{C_w|\psi\downarrow}$ by Lemma 4.2(2) because $\psi{\downarrow} \subseteq B$. Hence, by property (iii), $w \models \psi\langle\!\langle C_w \mid \psi{\downarrow}\rangle\!\rangle$ iff $w \models \psi$. Since $\psi \in C_w$, we have $w \models \psi$, and we are done.

($\Leftarrow$): Assume there is $C \subseteq B$ such that $w \models \varphi\langle C \,|\, B\rangle$ and $w \models \psi\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle$ for every $\psi \in C$. We prove $w \models \varphi$.

Claim. *$w \models \psi$ for every $\psi \in C$.*

Proof of the Claim. We proceed by induction on $(C, \prec)$. □

Basis. $\psi{\downarrow} = \emptyset$. Then $w \models \psi\langle\!\langle \emptyset \,|\, \emptyset\rangle\!\rangle$. By Lemma 4.2 and property (iii), ψ and $\psi\langle\!\langle \emptyset \,|\, \emptyset\rangle\!\rangle$ are equivalent, and so $w \models \psi$.
Step. $\psi{\downarrow} \neq \emptyset$. For every $\chi \in C \cap \psi{\downarrow}$, we have $\chi \prec \psi$ by definition of $\psi{\downarrow}$, so $w \models \chi$ by the induction hypothesis. Hence, $w \models \bigwedge_{\chi \in C \cap \psi\downarrow} \chi$, so $C \cap \psi{\downarrow} \subseteq C_w \cap \psi{\downarrow}$. Since $w \models \psi\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle$ by assumption, we get $w \models \psi\langle\!\langle C_w \,|\, \psi{\downarrow}\rangle\!\rangle$ by property (iv). Further, by property (iii), we have $w \models \psi\langle\!\langle C_w \,|\, \psi{\downarrow}\rangle\!\rangle$ iff $w \models \psi$. So $w \models \psi$.

By the claim, we have $w \models \bigwedge_{\psi \in C} \psi$ and so $C \subseteq C_w$. It follows $w \models \varphi\langle C_w \,|\, B\rangle$ by property (ii). By property (i), we have $w \models \varphi$ iff $w \models \varphi\langle C_w \,|\, B\rangle$, and so $w \models \varphi$. □

Remark 4.7. The Weak Contextual Equivalence Lemma is a corollary of the Contextual Equivalence Lemma. Indeed, the weak statement is obtained by choosing $\prec$ as the empty order (no two basis formulas are ordered), which is trivially well founded. This implies $\psi{\downarrow} = \emptyset$ and, by Lemma 4.2, we are forced to take $\psi\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle = \psi\langle\!\langle C \,|\, \emptyset\rangle\!\rangle = \psi$ for any $\psi \in B$ and context $C \subseteq \emptyset$.

In the rest of the section, we prove the Normalization Theorem by instantiating the Contextual Equivalence Lemma as follows:

- In Section 4.2, we define a well-founded basis $(B, \prec)$ for a given formula φ. Loosely speaking, B contains formulas of the form $\mathbf{GF}\psi$ and $\mathbf{FG}\psi$ for certain subformulas ψ of φ. Further, $\prec$ is induced by the subformula order: given basis formulas $op(\psi)$ and $op'(\psi')$, where $op, op' \in \{\mathbf{GF}, \mathbf{FG}\}$, we say $op(\psi) \prec op'(\psi')$ if ψ is a proper subformula of ψ'.
- In Section 4.3, we define formulas $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle$ and $(\mathbf{GF}\psi)\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle$ of Δ_2 for all basis formulas $\mathbf{FG}\psi, \mathbf{GF}\psi \in B$, and for every $C \subseteq B$. We prove that these formulas satisfy conditions (iii) and (iv) of the Contextual Equivalence Lemma.
- In Section 4.3.2, we define a formula $\varphi\langle C \,|\, B\rangle \in \Delta_2$ for every $C \subseteq B$, and prove that it satisfies conditions (i) and (ii) of the Contextual Equivalence Lemma.

The Contextual Equivalence Lemma yields that φ is equivalent to $\bigvee_{C \subseteq B} \left(\varphi\langle C \,|\, B\rangle \wedge \bigwedge_{\psi \in C} \psi\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle\right)$, a Boolean combination of formulas of Δ_2. Since Δ_2 is closed under Boolean combinations, we obtain a *contextual normalization* procedure.

4.2 Contextual Normalization I: The Well-Founded Basis $(B, \prec)$

In our introductory example at the beginning of Section 4.1, we normalize the formula $\mathbf{FG}(a \,\mathbf{U}\, b)$ with the help of a case distinction: for words satisfying $\mathbf{GF}b$, the formula is equivalent to $\mathbf{ff}$, and for words that do not satisfy $\mathbf{GF}b$, the formula is equivalent to $\mathbf{FG}(a \,\mathbf{U}\, b)$. This holds because b is the right child of the until operator; indeed, the case distinction with $\mathbf{GF}a$ does not work. Similarly, we normalize the formula $\mathbf{GF}(a \,\mathbf{W}\, b)$ by a case distinction on $\mathbf{FG}a$, and we succeed because a is the left child of the weak until. This motivates the following definition of a well-founded basis for a given formula.

Definition 4.8. Let φ be a formula, and let $sf(\varphi)$ denote the set of subformulas of φ. We define $B := B_{\mathrm{GF}} \cup B_{\mathrm{FG}}$, where $B_{\mathrm{GF}} := \{\mathbf{GF}\psi \mid \exists\chi.\ \chi \,\mathbf{U}\, \psi \in sf(\varphi) \vee \psi \,\mathbf{M}\, \chi \in sf(\varphi)\}$ and $B_{\mathrm{FG}} := \{\mathbf{FG}\psi \mid \exists\chi.\ \psi \,\mathbf{W}\, \chi \in sf(\varphi) \vee \chi \,\mathbf{R}\, \psi \in sf(\varphi)\}$. Further, we define a well-founded partial order $(\prec) \subseteq B \times B$ as follows: for every $op, op' \in \{\mathbf{GF}, \mathbf{FG}\}$, $op(\psi) \prec op'(\psi')$ iff $\psi \in sf(\psi')$.

Example 4.9. Consider the formula $\varphi = ((a \mathbf{W} b) \mathbf{U} c) \mathbf{W} d$. We have $B_{\mathbf{GF}} = \{\mathbf{GF}c\}$, $B_{\mathbf{FG}} = \{\mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)), \mathbf{FG}a\}$ and $B = \{\mathbf{GF}c, \mathbf{FG}((a \mathbf{W} b) \mathbf{U} c), \mathbf{FG}a\}$. Further, $\mathbf{GF}c \prec \mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)$ and $\mathbf{FG}a \prec \mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)$. Observe that the basis formula $\mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)$ is not in Δ_2.

4.3 Contextual Normalization II: The Formulas $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ and $(\mathbf{GF}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$

For all basis formulas $\mathbf{FG}\psi, \mathbf{GF}\psi \in B$ and for all $C \subseteq \psi\!\downarrow$, we define formulas $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ and $(\mathbf{GF}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ and prove that the definitions satisfies conditions (iii) and (iv) of the Contextual Equivalence Lemma. We define $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ and prove its properties in Section 4.3.1, and do the same for $(\mathbf{GF}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ in Section 4.3.2.

4.3.1 The Formula $\mathbf{FG}\psi\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$. Loosely speaking, we define $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ in two steps: first, we assign to ψ a formula $\psi\langle\!\langle C\rangle\!\rangle$ of Π_1, then set $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle := \mathbf{FG}(\psi\langle\!\langle C\rangle\!\rangle)$. Since $\psi\langle\!\langle C\rangle\!\rangle \in \Pi_1$, we have $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle \in \Delta_2$.

Definition 4.10. Let $\mathbf{FG}\psi \in B_{\mathbf{FG}}$ be a formula, and let $C \subseteq \psi\!\downarrow$. Further, let $\psi\langle\!\langle C\rangle\!\rangle$ be the formula inductively defined as follows. If $\psi = \mathbf{tt}, \mathbf{ff}, a, \neg a$, then $\psi\langle\!\langle C\rangle\!\rangle := \psi$. For the operators $\vee$, $\wedge$, $\mathbf{X}$, $\mathbf{W}$, and $\mathbf{R}$, the formula $\psi\langle\!\langle C\rangle\!\rangle$ is defined homomorphically.[1] For the operators $\mathbf{U}$ and $\mathbf{M}$, we set

$$(\psi_1 \mathbf{U} \psi_2)\langle\!\langle C\rangle\!\rangle := \begin{cases} \psi_1\langle\!\langle C\rangle\!\rangle \mathbf{W} \psi_2\langle\!\langle C\rangle\!\rangle & \text{if } \mathbf{GF}\psi_2 \in C \\ \mathbf{ff} & \text{otherwise,} \end{cases}$$

$$(\psi_1 \mathbf{M} \psi_2)\langle\!\langle C\rangle\!\rangle := \begin{cases} \psi_1\langle\!\langle C\rangle\!\rangle \mathbf{R} \psi_2\langle\!\langle C\rangle\!\rangle & \text{if } \mathbf{GF}\psi_1 \in C \\ \mathbf{ff} & \text{otherwise.} \end{cases}$$

We define $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle := \mathbf{FG}(\psi\langle\!\langle C\rangle\!\rangle)$.

Loosely speaking, in $\psi\langle\!\langle C\rangle\!\rangle$, all occurrences of $\mathbf{U}$ in ψ are either changed into $\mathbf{W}$ or removed. Therefore, $\psi\langle\!\langle C\rangle\!\rangle$ does not contain any occurrence of $\mathbf{U}$ anymore. The same happens with $\mathbf{M}$. So $\psi\langle\!\langle C\rangle\!\rangle$ is a formula of Π_1, and $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ is a formula of Δ_2.

Example 4.11. Consider the basis formula $\mathbf{FG}\psi = \mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)$ of Example 4.9. We have $\psi\!\downarrow = \{\mathbf{GF}c, \mathbf{FG}a\}$. For $C := \{\mathbf{GF}c\}$ and $C := \{\mathbf{GF}c, \mathbf{FG}a\}$, we get $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle = \mathbf{FG}((a \mathbf{W} b) \mathbf{W} c)$. For $C := \emptyset$ and $C := \{\mathbf{FG}a\}$, we get $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle = \mathbf{FGff} \equiv \mathbf{ff}$.

We prove in Proposition 4.14 in the following that $(\mathbf{FG}\psi)\langle\!\langle C \,|\, \psi\!\downarrow\rangle\!\rangle$ satisfies conditions (iii) and (iv) of the Contextual Equivalence Lemma. First we need a definition and a technical lemma.

Definition 4.12. Let $C \subseteq \psi\!\downarrow$ and $w \in L_{C|\psi\downarrow}$. The *stabilization index* of w with respect to C is the least index $I \geq 0$ such that $w_{I+j} \not\models \chi$ for every $j \geq 0$ and every formula $\mathbf{GF}\chi \in \psi\!\downarrow \setminus C$.

The stabilization index exists because, by the definition of $L_{C|\psi\downarrow}$, we have $w \not\models \mathbf{GF}\chi$ for every $\mathbf{GF}\chi \in \psi\!\downarrow \setminus C$, and so there exists only finitely many suffixes of w that satisfy each χ. Since $\psi\!\downarrow$ is finite, we can choose the stabilization index as the least index i such that no suffix w_{i+j} satisfies χ for any $\mathbf{GF}\chi \in \psi\!\downarrow \setminus C$. The following lemma explains the relation between ψ and $\psi\langle\!\langle C\rangle\!\rangle$.

Lemma 4.13. *Let $C \subseteq \psi\!\downarrow$, let $w \in L_{C|\psi\downarrow}$, and let I be the stabilization index of w with respect to C:*

(1) For every $i \geq I$, $w_i \models \psi \Rightarrow w_i \models \psi\langle\!\langle C\rangle\!\rangle$.
(2) For every $i \geq 0$, $w_i \models \psi\langle\!\langle C\rangle\!\rangle \Rightarrow w_i \models \psi$.

[1] Given an operator $\mathbf{P}$ of arity $k \geq 0$, we say that $\psi\langle C\rangle$ is defined *homomorphically* for $\mathbf{P}$ if $(\mathbf{P}(\psi_1, \ldots, \psi_k))\langle C\rangle = \mathbf{P}(\psi_1\langle C\rangle, \ldots, \psi_k\langle C\rangle)$. So, for example, $\mathbf{X}\psi_1\langle C\rangle = \mathbf{X}(\psi_1\langle C\rangle)$ and $(\psi_1 \wedge \psi_2)\langle C\rangle = \psi_1\langle C\rangle \wedge \psi_2\langle C\rangle$.

Proof. (1) Fix an $i \geq I$ such that $w_i \models \psi$. We prove $w_i \models \psi\langle\!\langle C \rangle\!\rangle$ by structural induction on ψ. If $\psi = \mathbf{tt}, \mathbf{ff}, a, \overline{a}$, then $\psi\langle\!\langle C \rangle\!\rangle = \psi$, and we are done. For all cases defined homomorphically, the result follows immediately from the induction hypothesis. Assume now $\psi = \psi_1 \mathbin{\mathbf{U}} \psi_2$. We claim $\mathbf{GF}\psi_2 \in C$. Observe first that, by the definition of the basis, we have $\mathbf{GF}\psi_2 \in \psi{\downarrow}$. Assume that $\mathbf{GF}\psi_2 \notin C$. Then, since $w \in L_{C|\psi\downarrow}$ and $i \geq I$, we have $w_{i+j} \not\models \psi_2$ for every $j \geq 0$, which implies $w_i \not\models \psi_1 \mathbin{\mathbf{U}} \psi_2$ by the semantics of LTL. But this contradicts the assumption $w_i \models \psi$, and the claim is proved. Now we proceed as follows:

$$\begin{array}{rcll}
w_i \models \psi_1 \mathbin{\mathbf{U}} \psi_2 & \Longrightarrow & w_i \models \psi_1 \mathbin{\mathbf{W}} \psi_2 & \text{(semantics of LTL)} \\
& \Longleftrightarrow & \forall j.\, (w_{i+j} \models \psi_1 \vee \exists k \leq j.\, w_{i+k} \models \psi_2) & \text{(semantics of LTL)} \\
& \Longrightarrow & \forall j.\, (w_{i+j} \models \psi_1\langle\!\langle C \rangle\!\rangle \vee \exists k \leq j.\, w_{i+k} \models \psi_2\langle\!\langle C \rangle\!\rangle) & \text{(induction hypothesis)} \\
& \Longrightarrow & w_i \models (\psi_1\langle\!\langle C \rangle\!\rangle) \mathbin{\mathbf{W}} (\psi_2\langle\!\langle C \rangle\!\rangle) & \text{(semantics of LTL)} \\
& \Longleftrightarrow & w_i \models (\psi_1 \mathbin{\mathbf{U}} \psi_2)\langle\!\langle C \rangle\!\rangle & \text{(definition of } \langle\!\langle\rangle\!\rangle \text{ and claim).}
\end{array}$$

Finally, assume $\psi = \psi_1 \mathbin{\mathbf{M}} \psi_2$. Using $\psi_1 \mathbin{\mathbf{M}} \psi_2 \equiv \psi_1 \mathbin{\mathbf{U}} (\psi_1 \wedge \psi_2)$, $\psi_1 \mathbin{\mathbf{R}} \psi_2 \equiv \psi_1 \mathbin{\mathbf{W}} (\psi_1 \wedge \psi_2)$, and the previous case, we get

$$\begin{aligned}
w_i \models \psi_1 \mathbin{\mathbf{M}} \psi_2 \iff w_i \models \psi_1 \mathbin{\mathbf{U}} (\psi_1 \wedge \psi_2) \implies w_i \models \psi_1\langle\!\langle C \rangle\!\rangle \mathbin{\mathbf{W}} (\psi_1\langle\!\langle C \rangle\!\rangle \wedge \psi_2\langle\!\langle C \rangle\!\rangle) \\
\iff w_i \models \psi_1\langle\!\langle C \rangle\!\rangle \mathbin{\mathbf{R}} \psi_2\langle\!\langle C \rangle\!\rangle \iff w_i \models (\psi_1 \mathbin{\mathbf{M}} \psi_2)\langle\!\langle C \rangle\!\rangle.
\end{aligned}$$

(2) Fix an $i \geq 0$ such that $w_i \models \psi\langle\!\langle C \rangle\!\rangle$. We prove $w_i \models \psi$ by structural induction on ψ. All cases but $\psi = \psi_1 \mathbin{\mathbf{U}} \psi_2$ are proved as in (1). Assume now $\varphi = \psi_1 \mathbin{\mathbf{U}} \psi_2$. By the definition of the basis, we have $\mathbf{GF}\psi_2 \in \psi{\downarrow}$. We claim $\mathbf{GF}\psi_2 \in C$. Assume the contrary. Then, by the definition of $\langle\!\langle\rangle\!\rangle$, we have $\psi\langle\!\langle C \rangle\!\rangle = \mathbf{ff}$, contradicting that $w_i \models \psi\langle\!\langle C \rangle\!\rangle$, and the claim is proved. Since $\mathbf{GF}\psi_2 \in C$ and $w \in L_{C|\psi\downarrow}$, we have $w \models \mathbf{GF}\psi_2$. We show $w_i \models \psi_1 \mathbin{\mathbf{U}} \psi_2$:

$$\begin{array}{rcll}
w_i \models (\psi_1 \mathbin{\mathbf{U}} \psi_2)\langle\!\langle C \rangle\!\rangle & \Longleftrightarrow & w_i \models (\psi_1\langle\!\langle C \rangle\!\rangle) \mathbin{\mathbf{W}} (\psi_2\langle\!\langle C \rangle\!\rangle) & \text{(definition of } \langle\!\langle\rangle\!\rangle \text{ and claim)} \\
& \Longleftrightarrow & \forall j.\, w_{i+j} \models \psi_1\langle\!\langle C \rangle\!\rangle \vee \exists k \leq j.\, w_{i+k} \models \psi_2\langle\!\langle C \rangle\!\rangle & \text{(semantics of LTL)} \\
& \Longrightarrow & \forall j.\, w_{i+j} \models \psi_1 \vee \exists k \leq j.\, w_{i+k} \models \psi_2 & \text{(induction hypothesis)} \\
& \Longleftrightarrow & w_i \models \psi_1 \mathbin{\mathbf{W}} \psi_2 & \text{(semantics of LTL)} \\
& \Longleftrightarrow & w_i \models \psi_1 \mathbin{\mathbf{U}} \psi_2 & \text{(because } w_i \models \mathbf{GF}\psi_2\text{).}
\end{array}$$

The case $\psi = \psi_1 \mathbin{\mathbf{M}} \psi_2$ is proved as in (1). □

Proposition 4.14. *Let* $\mathbf{FG}\psi$ *be a basis formula. For every* $C \subseteq \psi{\downarrow}$*, the formula* $\mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle)$ *belongs to* Δ_2*. Further,* $\mathbf{FG}\psi$ *and* $\mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle)$ *are equivalent under context* $\langle C \mid \psi{\downarrow} \rangle$*. Finally, if* $C \subseteq C'$*, then* $\mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle) \models \mathbf{FG}(\psi\langle\!\langle C' \rangle\!\rangle)$.

Proof. Fix $C \subseteq \psi{\downarrow}$. We prove the three parts of the proposition separately. (1) The formula $\mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle)$ belongs to Δ_2. Follows immediately from $\psi\langle\!\langle C \rangle\!\rangle \in \Pi_1$. (2) $\mathbf{FG}\psi$ and $\mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle)$ are equivalent with respect to $\langle C \mid \psi{\downarrow} \rangle$. If $L_{C|\psi\downarrow} = \emptyset$, both formulas are trivially equivalent, so let $w \in L_{C|\psi\downarrow}$. Let I be the stabilization index of w with respect to C. By Lemma 4.13, for every $i \geq 0$ we have $w_{i+I} \models \psi$ iff $w_{i+I} \models \psi\langle\!\langle C \rangle\!\rangle$. So, in particular, $w \models \mathbf{FG}\psi$ iff $w \models \mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle)$. (3) If $C \subseteq C'$, then $\mathbf{FG}(\psi\langle\!\langle C \rangle\!\rangle) \models \mathbf{FG}(\psi\langle\!\langle C' \rangle\!\rangle)$. Assume $C \subseteq C'$. By the monotonicity of $\mathbf{F}$ and $\mathbf{G}$, it suffices to prove $\psi\langle\!\langle C \rangle\!\rangle \models \psi\langle\!\langle C' \rangle\!\rangle$. We proceed by structural induction on ψ. The cases $\psi = \mathbf{tt}, \mathbf{ff}, a, \overline{a}$ are trivial. We now consider the case $\psi = \psi_1 \mathbin{\mathbf{W}} \psi_2$, as a representative of the cases where $\psi\langle\!\langle C \rangle\!\rangle$ is defined homomorphically. We have $\psi\langle\!\langle C \rangle\!\rangle = \psi_1\langle\!\langle C \rangle\!\rangle \mathbin{\mathbf{W}} \psi_2\langle\!\langle C \rangle\!\rangle$ and $\psi\langle\!\langle C' \rangle\!\rangle = \psi_1\langle\!\langle C' \rangle\!\rangle \mathbin{\mathbf{W}} \psi_2\langle\!\langle C' \rangle\!\rangle$ (definition of $\langle\!\langle C \rangle\!\rangle$). So it suffices to show $\psi_1\langle\!\langle C \rangle\!\rangle \mathbin{\mathbf{W}} \psi_2\langle\!\langle C \rangle\!\rangle \models \psi_1\langle\!\langle C' \rangle\!\rangle \mathbin{\mathbf{W}} \psi_2\langle\!\langle C' \rangle\!\rangle$. But this follows immediately from $\psi_1\langle\!\langle C \rangle\!\rangle \models \psi_1\langle\!\langle C' \rangle\!\rangle$ and $\psi_2\langle\!\langle C \rangle\!\rangle \models \psi_2\langle\!\langle C' \rangle\!\rangle$ (the induction hypothesis), and the monotonicity of $\mathbf{W}$. Finally, we consider the two remaining cases.

Case $\psi = \psi_1 \mathbin{\mathbf{U}} \psi_2$. We have $\mathbf{GF}\psi_2 \in \psi{\downarrow}$ (definition of $\psi{\downarrow}$). If $\mathbf{GF}\psi_2 \notin C$, then $\psi\langle\!\langle C \rangle\!\rangle = \mathbf{ff}$ (definition of $\langle\!\langle C \rangle\!\rangle$), which implies $\psi\langle\!\langle C \rangle\!\rangle \models \psi\langle\!\langle C' \rangle\!\rangle$, and we are done. If $\mathbf{GF}\psi_2 \in C$, then $\mathbf{GF}\psi_2 \in C'$ ($C \subseteq C'$), and so we have $\psi\langle\!\langle C\rangle\!\rangle = \psi_1\langle\!\langle C\rangle\!\rangle \mathbf{W} \psi_2\langle\!\langle C\rangle\!\rangle$ and $\psi\langle\!\langle C'\rangle\!\rangle = \psi_1\langle\!\langle C'\rangle\!\rangle \mathbf{W} \psi_2\langle\!\langle C'\rangle\!\rangle$ (definition of $\langle\!\langle C\rangle\!\rangle$). Proceed now as in the previous case.

Case $\psi = \psi_1 \mathbf{M} \psi_2$. We have $\mathbf{GF}\psi_1 \in \psi{\downarrow}$ (definition of $\psi{\downarrow}$). If $\mathbf{GF}\psi_1 \notin C$, then we proceed as for $\varphi = \psi_1 \mathbf{U} \psi_2$. If $\mathbf{GF}\psi_1 \in C$, then $\mathbf{GF}\psi_1 \in C'$ ($C \subseteq C'$). Proceed as in the previous case, replacing $\mathbf{W}$ by $\mathbf{R}$. □

4.3.2 *The Formula* $\mathbf{GF}\psi\langle\!\langle C \mid \psi{\downarrow}\rangle\!\rangle$. We define the formula $(\mathbf{GF}\psi)\langle\!\langle C \mid \psi{\downarrow}\rangle\!\rangle$ and prove that it satisfies conditions (iii) and (iv) of the Contextual Equivalence Lemma. We use the duality between $\mathbf{FG}$ and $\mathbf{GF}$. Recall from Section 2.1 that the dual $\overline{\varphi}$ of a formula φ in negation normal form replaces $\mathbf{U}$ by $\mathbf{R}$, $\mathbf{W}$ by $\mathbf{M}$, and vice versa. In particular, we have $\overline{\mathbf{FG}\varphi} = \mathbf{GF}\,\overline{\varphi}$ and $\overline{\mathbf{GF}\varphi} = \mathbf{FG}\,\overline{\varphi}$, $\overline{\Pi_1} = \Sigma_1$, and $\overline{\Delta_2} = \Delta_2$ by Proposition 3.3. Moreover, by Lemma 4.2(4), two formulas are equivalent under context $\langle C_d \mid \overline{B}\rangle$ iff they are equivalent under the dual context $\langle C \mid B\rangle$. This is all we need.

Definition 4.15. Let $\mathbf{GF}\psi \in B$, and let $C \subseteq \psi{\downarrow}$. We define $\psi[\![C]\!] \coloneqq \overline{\overline{\psi}\langle\!\langle C_d\rangle\!\rangle}$ and $(\mathbf{GF}\psi)\langle\!\langle C \mid \psi{\downarrow}\rangle\!\rangle \coloneqq \mathbf{GF}(\psi[\![C]\!])$.

Since $\overline{\psi}\langle\!\langle C_d\rangle\!\rangle \in \Pi_1$, we have $\psi[\![C]\!] \in \overline{\Pi_1} = \Sigma_1$. We now prove the counterpart of Proposition 4.14.

Proposition 4.16. *Let* $\mathbf{GF}\,\psi$ *be a basis formula. For every* $C \subseteq \psi{\downarrow}$, *the formula* $\mathbf{GF}\left(\psi[\![C]\!]\right)$ *belongs to* Δ_2. *Further,* $\mathbf{GF}\,\psi$ *and* $\mathbf{GF}\left(\psi[\![C]\!]\right)$ *are equivalent under context* $\langle C \mid \psi{\downarrow}\rangle$. *Finally, if* $C \subseteq C'$, *then* $\mathbf{GF}\left(\psi[\![C]\!]\right) \models \mathbf{GF}(\psi[\![C']\!])$.

Proof. Each of the three statements follows by duality from Proposition 4.14. Given any formula $\mathbf{GF}\psi$ and any two sets $C, C' \subseteq \psi{\downarrow}$, we instantiate Proposition 4.14 with $\overline{\mathbf{GF}\psi}$, C_d, and C'_d. For the first statement, we have $\mathbf{GF}(\psi[\![C]\!]) = \overline{\mathbf{FG}(\overline{\psi}\langle\!\langle C_d\rangle\!\rangle)} \in \overline{\Delta_2} = \Delta_2$. Let us now prove that $\mathbf{GF}\,\psi$ and $\mathbf{GF}\left(\psi[\![C]\!]\right)$ are equivalent under context $\langle C \mid \psi{\downarrow}\rangle$. We have $\mathbf{GF}\psi = \overline{\mathbf{FG}\overline{\psi}}$ and $\mathbf{GF}(\psi[\![C]\!]) = \overline{\mathbf{FG}(\overline{\psi}\langle\!\langle C_d\rangle\!\rangle)}$ by definition. So it suffices to show that $\mathbf{FG}\overline{\psi}$ and $\mathbf{FG}(\overline{\psi}\langle\!\langle C_d\rangle\!\rangle)$ are equivalent under $\langle C \mid \psi{\downarrow}\rangle$, and, by Lemma 4.2(4), that they are equivalent under context $\langle C_d \mid \overline{\psi}{\downarrow}\rangle$. But this follows from Proposition 4.16. Finally, let $C \subseteq C'$. We show $\mathbf{GF}\left(\psi[\![C]\!]\right) \models \mathbf{GF}(\psi[\![C']\!])$. Since $C \subseteq C'$, we have $C'_d \subseteq C_d$ and so $\neg\,\mathbf{GF}(\psi[\![C'_d]\!]) \equiv \mathbf{FG}(\overline{\psi}\langle\!\langle C'_d\rangle\!\rangle) \models \mathbf{FG}(\overline{\psi}\langle\!\langle C_d\rangle\!\rangle) \equiv \overline{\mathbf{GF}(\psi[\![C_d]\!])} \equiv \neg\mathbf{GF}(\psi[\![C_d]\!])$, where the $\models$-step applies Proposition 4.14. By *modus tollens*, we are done. □

4.4 Contextual Normalization III: The Formula $\varphi\langle C \mid B\rangle$

We search for a formula $\varphi\langle C \mid B\rangle$ that satisfies conditions (i) and (ii) of the Contextual Equivalence Lemma, and belongs to Δ_2. Lemma 4.17 gives such a formula for the case in which $\varphi = \mathbf{G}\psi$ for some formula ψ. Let us first explain the idea. By the equivalence $\mathbf{G}\psi \equiv \psi \mathbf{U} \mathbf{G}\psi$, given a word w and $C \subseteq B$, the following is true for *every* index $j \geq 0$: $w \models \psi \mathbf{U} \mathbf{G}\psi$ iff $w_i \models \psi$ for every $i \leq j$ and $w_j \models \mathbf{G}\psi$. Observe that the form of this statement is already close to condition (i) of the Contextual Equivalence Lemma. The final trick is to choose j as the stabilization index I of w with respect to C. Applying Lemma 4.13, we obtain that $\mathbf{G}\psi$ is equivalent to $\psi \mathbf{U} \mathbf{G}\left(\psi\langle\!\langle C\rangle\!\rangle\right)$ under context $\langle C \mid B\rangle$, and also to $(\mathbf{G}\left(\psi\langle\!\langle C\rangle\!\rangle\right)) \mathbf{R}\, \psi$.

Lemma 4.17. *If* $\varphi = \mathbf{G}\psi$, *then for every* $C \subseteq B$ *the formulas* $\mathbf{G}\psi$, $\psi \mathbf{U} \mathbf{G}\left(\psi\langle\!\langle C\rangle\!\rangle\right)$, *and* $\mathbf{G}\left(\psi\langle\!\langle C\rangle\!\rangle\right) \mathbf{R}\, \psi$ *are equivalent under context* $\langle C \mid B\rangle$.

Proof. By Lemma 4.2(3), it suffices to show that they are equivalent under context $\langle C \mid B_{\mathrm{GF}}\rangle$ since $B_{\mathrm{GF}} \subseteq B$. We prove $w \models \mathbf{G}\psi$ iff $w \models \psi \mathbf{U} \mathbf{G}\left(\psi\langle\!\langle C\rangle\!\rangle\right)$ iff $w \models \mathbf{G}\left(\psi\langle\!\langle C\rangle\!\rangle\right) \mathbf{R} \psi$ for every $w \in L_{C|B_{\mathrm{GF}}}$. So fix an arbitrary $w \in L_{C|B_{\mathrm{GF}}}$.

($\Rightarrow$) Assume $w \models \mathbf{G}\psi$. Then, in particular, we have $w \models \mathbf{FG}\psi$. By Lemma 4.13, ($w_i \models \psi \Leftrightarrow \psi\langle\!\langle C\rangle\!\rangle$) holds for every i beyond the stabilization index of w with respect to C. So, in particular, we have $w \models \mathbf{FG}\,(\psi\langle\!\langle C\rangle\!\rangle)$. It follows that $w \models \mathbf{G}\psi \wedge \mathbf{FG}\,(\psi\langle\!\langle C\rangle\!\rangle)$, which, by the semantics of LTL, implies $w \models \psi\,\mathbf{U}\,\mathbf{G}\,(\psi\langle\!\langle C\rangle\!\rangle)$ and $w \models \mathbf{G}\,(\psi\langle\!\langle C\rangle\!\rangle)\,\mathbf{R}\,\psi$.

($\Leftarrow$) Assume $w \models \psi\,\mathbf{U}\,\mathbf{G}\,(\psi\langle\!\langle C\rangle\!\rangle)$. By the semantics of LTL, there exists $i \geq 0$ such that $w_j \models \psi$ for every $j < i$ and $w_j \models \psi\langle\!\langle C\rangle\!\rangle$ for every $j \geq i$. By Lemma 4.13, since $w \in L_{C|B_{\mathbf{GF}}}$, we have $w_j \models \psi$ for every $j \geq i$. So $w_j \models \psi$ holds for every $j \geq 0$, and so $w \models \mathbf{G}\psi$. For $\mathbf{G}\,(\psi\langle\!\langle C\rangle\!\rangle)\,\mathbf{R}\,\psi \equiv \mathbf{G}\,(\psi\langle\!\langle C\rangle\!\rangle)\,\mathbf{M}\,\psi \vee \mathbf{G}\psi$, either $\mathbf{G}\psi$ holds and we are already done or we can apply the previous argument with only $j \leq i$ instead of $j < i$. □

It is easy to extend Lemma 4.17 to all **W**- and **R**-formulas.

Proposition 4.18. *Let ψ_1 and ψ_2 be formulas:*

(1) For every $C \subseteq B_{\mathbf{GF}}$, the formulas $\psi_1\,\mathbf{W}\,\psi_2$ and $\psi_1\,\mathbf{U}\,(\psi_2 \vee \mathbf{G}\,(\psi_1\langle\!\langle C\rangle\!\rangle))$ are equivalent under context C.

(2) For every $C \subseteq B_{\mathbf{GF}}$, the formulas $\psi_1\,\mathbf{R}\,\psi_2$ and $(\psi_1 \vee \mathbf{G}\,(\psi_2\langle\!\langle C\rangle\!\rangle))\,\mathbf{M}\,\psi_2$ are equivalent under context C.

Proof. We only prove the first case. If $L_{C|B_{\mathbf{GF}}} = \emptyset$, the proposition trivially holds. Let $w \in L_{C|B_{\mathbf{GF}}}$.

($\Rightarrow$) Assume $w \models \psi_1\,\mathbf{W}\,\psi_2$, and consider two cases. If $w \models \mathbf{G}\psi_1$, then $w \models \psi_1\,\mathbf{U}\,\mathbf{G}(\psi_1\langle\!\langle C\rangle\!\rangle)$ (Lemma 4.17), and so $w \models \psi_1\,\mathbf{U}\,(\psi_2 \vee \mathbf{G}\,(\psi_1\langle\!\langle C\rangle\!\rangle))$. If $w \not\models \mathbf{G}\psi_1$, then we have $w \models \psi_1\,\mathbf{W}\,\psi_2$ iff $w \models \psi_1\,\mathbf{U}\,\psi_2$ by the semantics of LTL, and then $w \models \psi_1\,\mathbf{U}\,\psi_2$ implies $w \models \psi_1\,\mathbf{U}\,(\psi_2 \vee \mathbf{G}\,(\psi_1\langle\!\langle C\rangle\!\rangle))$, also by the semantics of LTL.

($\Leftarrow$) Assume $w \models \psi_1\,\mathbf{U}\,(\psi_2 \vee \mathbf{G}\,(\psi_1\langle\!\langle C\rangle\!\rangle))$. Then $w \models \psi_1\,\mathbf{U}\,(\psi_2 \vee \mathbf{G}\psi_1)$ (Lemma 4.17, after distributing **U** over $\vee$), and since $\psi_1\,\mathbf{U}\,(\psi_2 \vee \mathbf{G}\psi_1) \equiv \psi_1\,\mathbf{W}\,\psi_2$, we are done. □

Proposition 4.18 suggests the following definition for the formula $\varphi\langle C\rangle$.

Definition 4.19. For every $C \subseteq B$, the formula $\varphi\langle C\rangle$ is inductively defined as follows. If $\varphi = \mathbf{tt}, \mathbf{ff}, a, \overline{a}$, then $\varphi\langle C\rangle = \varphi$. For the operators $\vee$, $\wedge$, **X**, **U**, and **M**, we define $\varphi\langle C\rangle$ homomorphically. Finally,

$$(\psi_1\,\mathbf{W}\,\psi_2)\langle C\rangle := \psi_1\langle C\rangle\,\mathbf{U}\,\big(\psi_2\langle C\rangle \vee \mathbf{G}\,(\psi_1\langle\!\langle C\rangle\!\rangle)\big) \quad \text{and} \quad (\psi_1\,\mathbf{R}\,\psi_2)\langle C\rangle = \big(\psi_1\langle C\rangle \vee \mathbf{G}(\psi_2\langle\!\langle C\rangle\!\rangle)\big)\,\mathbf{M}\,\psi_2\langle C\rangle\,.$$

Example 4.20. Let $\varphi = ((a\,\mathbf{W}\,b)\,\mathbf{U}\,c)\,\mathbf{W}\,d$ and $C \subseteq B$. With $((a\,\mathbf{W}\,b)\,\mathbf{U}\,c)\langle\!\langle C\rangle\!\rangle = (a\,\mathbf{W}\,b)\,\mathbf{W}\,c$, we get

$$\begin{aligned}\varphi\langle C\rangle &= \big((a\,\mathbf{W}\,b)\,\mathbf{U}\,c\big)\langle C\rangle\ \mathbf{U}\ \big(d \vee \mathbf{G}\big((a\,\mathbf{W}\,b)\,\mathbf{W}\,c\big)\big)\\ &= \big((a\,\mathbf{W}\,b)\langle C\rangle\,\mathbf{U}\,c\big)\ \mathbf{U}\ \big(d \vee \mathbf{G}\big((a\,\mathbf{W}\,b)\,\mathbf{W}\,c\big)\big)\\ &= \big(\big(a\,\mathbf{U}\,(b \vee \mathbf{G}a)\big)\,\mathbf{U}\,c\big)\ \mathbf{U}\ \big(d \vee \mathbf{G}\big((a\,\mathbf{W}\,b)\,\mathbf{W}\,c\big)\big)\,.\end{aligned}$$

We prove that setting $\varphi\langle C \mid B\rangle := \varphi\langle C\rangle$ satisfies all conditions of the Contextual Equivalence Lemma.

Proposition 4.21. *Let φ be a formula. For every $C \subseteq B$, the formula $\varphi\langle C\rangle$ belongs to Σ_2. Further, φ and $\varphi\langle C\rangle$ are equivalent under context $\langle C\,|\,B\rangle$. Finally, if $C \subseteq C'$, then $\varphi\langle C\rangle \models \varphi\langle C'\rangle$.*

Proof. (1) The formula $\varphi\langle C\rangle$ belongs to Σ_2. We show it by structural induction on φ. The only non-trivial cases are $\varphi = \varphi_1\,\mathbf{W}\,\varphi_2$ and $\varphi = \varphi_1\,\mathbf{R}\,\varphi_2$. For either of the two cases, by the induction hypothesis, we have $\varphi_k\langle C\rangle \in \Sigma_2$ for $k = 1, 2$. Further, since $\varphi_k\langle\!\langle C\rangle\!\rangle \in \Pi_1$, we have $\mathbf{G}\,(\varphi_k\langle\!\langle C\rangle\!\rangle) \in \Pi_1$ as well. So $\varphi_1\langle C\rangle\,\mathbf{U}\,\big(\varphi_2\langle C\rangle \vee \mathbf{G}\,(\varphi_1\langle\!\langle C\rangle\!\rangle)\big) \in \Sigma_2$ and $\big(\varphi_1\langle C\rangle \vee \mathbf{G}\big(\varphi_2\langle\!\langle C\rangle\!\rangle\big)\big)\,\mathbf{M}\,\varphi_2\langle C\rangle \in \Sigma_2$, and we are done.

(2) The formulas φ and $\varphi\langle C\rangle$ are equivalent under context $\langle C \mid B\rangle$. By Lemma 4.2(3), it suffices to show that $(w \models \varphi \iff w \models \varphi\langle C\rangle)$ holds for every $C \subseteq B_{\mathbf{GF}}$ and for every $w \in L_{C|B_{\mathbf{GF}}}$. We proceed by structural induction on φ. We use the identity

$$\psi\langle C\rangle = \psi\langle C \cap B_{\mathbf{GF}}\rangle,$$

which holds because the case distinctions in Definition 4.10 only involve formulas $\mathbf{GF}\chi$ such that χ is a subformula of ψ, and those formulas belong to $B_{\mathbf{GF}}$.

The base of the induction is $\varphi \in \{\mathbf{tt}, \mathbf{ff}, a, \neg a\}$. In all of these cases, we have $\varphi = \varphi\langle C\rangle$ by definition, and we are done. All cases in which $\varphi\langle C\rangle$ is defined homomorphically are handled in the same way, and so we consider only the case $\varphi = \psi_1 \mathbf{U}\, \psi_2$. Fix $C \subseteq B_{\mathbf{GF}}$ and $w \in L_{C|B_{\mathbf{GF}}}$. We prove $(w \models \varphi \iff w \models \varphi\langle C\rangle)$. Applying the induction hypothesis to ψ_1 and ψ_2, ψ_j and $\psi_j\langle C\rangle$ are equivalent under context $\langle C \mid B_{\mathbf{GF}}\rangle$. Moreover, $w_i \in L_{C|B_{\mathbf{GF}}}$ for all $i \geq 0$ by the limit properties of $\mathbf{GF}$-subformulas, and this yields

$$\forall i.(w_i \models \psi_j \iff w_i \models \psi_j\langle C\rangle) \qquad \text{for } j = 1, 2. \tag{2}$$

We get

$$\begin{aligned} w \models \psi_1 \mathbf{U}\, \psi_2 &\iff \exists k.\ w_k \models \psi_2 \wedge (\forall \ell < k.\ w_\ell \models \psi_1) && \text{(semantics of LTL)}\\ &\iff \exists k.\ w_k \models \psi_2\langle C\rangle \wedge (\forall \ell < k.\ w_\ell \models \psi_1\langle C\rangle) && \text{(2)}\\ &\iff w \models \psi_1\langle C\rangle \,\mathbf{U}\, \psi_2\langle C\rangle && \text{(semantics of LTL)}\\ &\iff w \models (\psi_1 \mathbf{U}\, \psi_2)\langle C\rangle && \text{(definition of } \langle\rangle\text{)}, \end{aligned}$$

which concludes the proof.

The remaining cases are $\varphi = \psi_1 \mathbf{W}\, \psi_2$ and $\varphi = \psi_1 \mathbf{R}\, \psi_2$. Fix $C \subseteq B_{\mathbf{GF}}$ and $w \in L_{C|B_{\mathbf{GF}}}$. By the induction hypothesis, (2) holds. We derive

$$\begin{aligned} w \models \psi_1 \mathbf{W}\, \psi_2 &\iff w \models \psi_1 \mathbf{U}\, (\psi_2 \vee \mathbf{G}(\psi_1\langle\!\langle C\rangle\!\rangle)) && \text{(Proposition 4.18)}\\ &\iff w \models\ \psi_1\langle C\rangle \,\mathbf{U}\, (\psi_2\langle C\rangle \vee \mathbf{G}(\psi_1\langle\!\langle C\rangle\!\rangle)) && \text{(2)}\\ &\iff w \models\ (\psi_1 \mathbf{W}\, \psi_2)\langle C\rangle && \text{(definition of } \langle\rangle\text{)}. \end{aligned}$$

The case $\varphi = \psi_1 \mathbf{R}\, \psi_2$ is analogous.

(3) If $C \subseteq C'$, then $\varphi\langle C\rangle \models \varphi\langle C'\rangle$. Follows from a straightforward structural induction, using $\varphi\langle\!\langle C\rangle\!\rangle \models \varphi\langle\!\langle C'\rangle\!\rangle$, and the monotonicity of all operators (recall that formulas are in negation normal form). □

4.5 Contextual Normalization IV: The Normalization Theorem

We insert Propositions 4.14, 4.16, and 4.21 into the Contextual Equivalence Lemma. This yields a closed expression for a formula of Δ_2 equivalent to a given formula. The normalized formula is exponentially larger than the original one.

Theorem 4.22 (Normalization Theorem). *Let φ be a formula of length n. We have*

$$\varphi \equiv \bigvee_{\substack{M \subseteq B_{\mathbf{GF}}\\ N \subseteq B_{\mathbf{FG}}}} \left(\varphi\langle M\rangle \wedge \bigwedge_{\mathbf{FG}\psi \in N} \mathbf{FG}(\psi\langle\!\langle M\rangle\!\rangle) \wedge \bigwedge_{\mathbf{GF}\psi \in M} \mathbf{GF}(\psi[\![N]\!]) \right).$$

Moreover, the right-hand-side formula is in Δ_2 and has length $2^{O(n)}$.

Proof. By the Contextual Equivalence Lemma, we have

$$\varphi \equiv \bigvee_{C \subseteq B} \left(\varphi\langle C \,|\, B\rangle \wedge \bigwedge_{\psi \in C} \psi\langle\!\langle C \,|\, \psi{\downarrow}\rangle\!\rangle \right)$$

for any well-founded basis $(B, \prec)$, formulas $\varphi\langle C\,|\,B\rangle$ satisfying conditions (i) and (ii), and formulas $\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle$ satisfying conditions (iii) and (iv) of the lemma.

We choose $(B, \prec)$ as in Definition 4.1. We have $B = B_{\mathbf{GF}} \uplus B_{\mathbf{FG}}$. For every $C \subseteq B$, we split $C = M \uplus N$, where $M := C \cap B_{\mathbf{GF}}$ and $N := C \cap B_{\mathbf{FG}}$. For every $\mathbf{FG}\psi \in N$ (recall that N only contains $\mathbf{FG}$-formulas) and $C \subseteq \psi{\downarrow}$, we define $(\mathbf{FG}\psi)\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle := \mathbf{FG}(\psi\langle\!\langle M\rangle\!\rangle)$, and for every $\mathbf{GF}\psi \in M$ and $C \subseteq \psi{\downarrow}$, we define $(\mathbf{GF}\psi)\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle := \mathbf{GF}(\psi[\![N]\!])$, as in Definitions 4.10 and 4.15. By Propositions 4.14 and 4.16, $\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle$ satisfies conditions (iii) and (iv) of the Contextual Equivalence Lemma for every $\psi \in B$ and $C \subseteq \psi{\downarrow}$. Finally, for every $C \subseteq B$, we define $\varphi\langle C\,|\,B\rangle := \varphi\langle C\rangle$, as in Definition 4.19. By Proposition 4.21, $\varphi\langle C\,|\,B\rangle$ satisfies conditions (i) and (ii) of the Contextual Equivalence Lemma. Applying the lemma, we obtain the following:

$$\begin{aligned}\varphi \equiv \bigvee_{C\subseteq B}\left(\varphi\langle C\,|\,B\rangle \wedge \bigwedge_{\psi\in C}\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle\right) &\equiv \bigvee_{\substack{M\subseteq B_{\mathbf{GF}}\\ N\subseteq B_{\mathbf{FG}}}}\left(\varphi\langle C\,|\,B\rangle \wedge \bigwedge_{\psi\in N}\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle \wedge \bigwedge_{\psi\in M}\psi\langle\!\langle C\,|\,\psi{\downarrow}\rangle\!\rangle\right)\\ &\equiv \bigvee_{\substack{M\subseteq B_{\mathbf{GF}}\\ N\subseteq B_{\mathbf{FG}}}}\left(\varphi\langle M\rangle \wedge \bigwedge_{\mathbf{FG}\psi\in N}\mathbf{FG}(\psi\langle\!\langle M\rangle\!\rangle) \wedge \bigwedge_{\mathbf{GF}\psi\in M}\mathbf{GF}(\psi[\![N]\!])\right).\end{aligned}$$

By Propositions 4.14, 4.16, and 4.21, the right-hand side is a Boolean combination of formulas of Δ_2, and so in Δ_2 itself. For the size, observe first that B contains at most n formulas, and so the right-hand side is a disjunction of at most 2^n formulas. We bound the length of a disjunct. First we observe that $|\varphi\langle C\rangle| \leq O(n^2)$, which follows easily from Definition 4.19. (For the cases defined homomorphically, the proof is a direct application of the induction hypothesis; for the case $\varphi = \psi_1 \mathbf{W} \psi_2$, we have $|\varphi\langle C\rangle| \leq |\psi_1\langle C\rangle| + |\psi_2\langle C\rangle| + |\psi_2\langle\!\langle C\rangle\!\rangle| + 2$, and the result follows from $|\psi_k\langle\!\langle C\rangle\!\rangle| \leq n$; the case $\varphi = \psi_1 \mathbf{R} \psi_2$ is analogous.) From the definition of $\varphi\langle\!\langle C\rangle\!\rangle$, we immediately obtain $|\mathbf{FG}(\varphi\langle\!\langle C\rangle\!\rangle)| \leq n$, and, since for every formula ψ we have $|\psi| = |\overline{\psi}|$, the same holds for $\mathbf{FG}(\varphi\langle\!\langle C\rangle\!\rangle)$. So each disjunct has length at most $O(n^2)+n\cdot n \in O(n^2)$. So we obtain $2^n \cdot O(n^2) \in 2^{O(n)}$ for the final bound. □

Example 4.23. Consider the formula $\varphi = ((a \mathbf{W} b) \mathbf{U} c) \mathbf{W} d$ of Example 4.9. We have $B_{\mathbf{GF}} = \{\mathbf{GF}c\}$ and $B_{\mathbf{FG}} = \{\mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)), \mathbf{FG}a\}$. Let us compute the disjunct of the right-hand side of Theorem 4.22 for $M := \{\mathbf{GF}c\}$ and $N := \{\mathbf{FG}((a \mathbf{W} b) \mathbf{U} c)), \mathbf{FG}a\}$. We have the following:

- $\varphi\langle M\rangle = ((a \mathbf{U} (b \vee \mathbf{G}a)) \mathbf{U} c) \mathbf{U} (d \vee \mathbf{G}((a \mathbf{W} b) \mathbf{W} c)) \in \Delta_2$, as shown in Example 4.20.
- $\mathbf{FG}(((a \mathbf{W} b) \mathbf{U} c)\langle\!\langle M\rangle\!\rangle) = \mathbf{FG}((a \mathbf{W} b) \mathbf{W} c) \in \Delta_2$, as shown in Example 4.11, and $\mathbf{FG}(a\langle\!\langle M\rangle\!\rangle) = \mathbf{FG}a$.
- $\mathbf{GF}(c[\![N]\!]) = \mathbf{GF}c$.

So the disjunct is $((a \mathbf{U} (b \vee \mathbf{G}a)) \mathbf{U} c) \mathbf{U} (d \vee \mathbf{G}((a \mathbf{W} b) \mathbf{W} c)) \wedge \mathbf{FG}((a \mathbf{W} b) \mathbf{W} c)) \wedge \mathbf{FG}\,a \wedge \mathbf{GF}\,c$.

We conclude with the following lemma that bounds the number of formulas in the normal form of Theorem 4.22. It will be used later in Section 6.3 to bound the size of some automata.

Lemma 4.24. *For any formula φ, $M \subseteq B_{\mathbf{GF}}$, and $N \subseteq B_{\mathbf{FG}}$, let $\varphi_{M,N}$ be the corresponding clause of the disjunction in Theorem 4.22. Then $|sf(\varphi_{M,N})| \in O(|sf(\varphi)|)$.*

Proof. This follows from the following claims:

(1) $|\bigcup\{sf(\psi\langle\!\langle M\rangle\!\rangle) : \psi \in sf(\varphi)\}| \leq |sf(\varphi)|$ and $|\bigcup\{sf(\psi[\![N]\!]) : \psi \in sf(\varphi)\}| \leq |sf(\varphi)|$
(2) $|sf(\varphi\langle M\rangle)| \leq 4|sf(\varphi)|$.

To prove (1), observe that $sf(\psi\langle\!\langle M\rangle\!\rangle) \subseteq \{\psi'\langle\!\langle M\rangle\!\rangle \mid \psi' \in sf(\psi)\}$. This can be proven by a straightforward induction, which is trivial for the cases where $\psi\langle\!\langle M\rangle\!\rangle$ is defined homomorphically. For

$\psi = \psi_1 \mathbf{U} \psi_2$, according to Definition 4.10, either $\psi\langle\!\langle M\rangle\!\rangle = \mathbf{ff}$ or $\psi\langle\!\langle M\rangle\!\rangle = \psi_1\langle\!\langle M\rangle\!\rangle \mathbf{W} \psi_2\langle\!\langle M\rangle\!\rangle$. Since $\psi\langle\!\langle M\rangle\!\rangle$ is in the right-hand set by definition with $\psi' = \psi$, it remains to consider the subformulas of $\psi_k\langle\!\langle M\rangle\!\rangle$ for $k = 1, 2$. However, by the induction hypothesis, for any $\psi'' \in sf(\psi_k\langle\!\langle M\rangle\!\rangle)$, there is a $\psi' \in sf(\psi_k) \subseteq sf(\psi)$ such that $\psi'' = \psi'\langle\!\langle M\rangle\!\rangle$, and we are done. Using the just proven inclusion, we have

$$\bigcup\{sf(\psi\langle\!\langle M\rangle\!\rangle) : \psi \in sf(\varphi)\} \subseteq \{\psi'\langle\!\langle M\rangle\!\rangle : \psi' \in sf(\psi), \psi \in sf(\varphi)\} = \{\psi\langle\!\langle M\rangle\!\rangle : \psi \in sf(\varphi)\},$$

and the cardinality of the latter set is clearly no more than $|sf(\varphi)|$. The case of $\psi[\![N]\!]$ follows by duality.

For (2), we prove the stronger claim $|sf(\psi\langle M\rangle) \cup sf(\psi\langle\!\langle M\rangle\!\rangle)| \leq 4|sf(\psi)|$ by induction on ψ. The base cases and those in which both $\psi\langle M\rangle$ and $\psi\langle\!\langle M\rangle\!\rangle$ are defined homomorphically are straightforward. Let us show only the cases of **U** and **W**, since **M** and **R** are completely analogous. For $\psi = \psi_1 \mathbf{U} \psi_2$, we have $\psi\langle M\rangle = \psi_1\langle M\rangle \mathbf{U} \psi_2\langle M\rangle$ and either $\psi\langle\!\langle M\rangle\!\rangle = \mathbf{ff}$ or $\psi\langle\!\langle M\rangle\!\rangle = \psi_1\langle\!\langle M\rangle\!\rangle \mathbf{W} \psi_2\langle\!\langle M\rangle\!\rangle$. In the first case, by the induction hypothesis, $|sf(\psi_k\langle M\rangle)| \leq |sf(\psi_k\langle M\rangle) \cup sf(\psi_k\langle\!\langle M\rangle\!\rangle)| \leq 4|sf(\psi_k)|$ for $k = 1, 2$. The remaining subformulas in the left-hand side are only two, **ff** and $\psi\langle M\rangle$ itself, so they are at most $4|sf(\psi)|$. In the second case,

$$sf(\psi\langle M\rangle) \cup sf(\psi\langle\!\langle M\rangle\!\rangle) = (sf(\psi_1\langle M\rangle) \cup sf(\psi_1\langle\!\langle M\rangle\!\rangle)) \cup (sf(\psi_2\langle M\rangle) \cup sf(\psi_2\langle\!\langle M\rangle\!\rangle)) \cup \{\psi\langle M\rangle, \psi\langle\!\langle M\rangle\!\rangle\},$$

so we have the bound $4|sf(\psi_1)| + 4|sf(\psi_2)| + 2 \leq 4|sf(\psi)|$ by the induction hypothesis. For $\psi = \psi_1 \mathbf{W} \psi_2$, we have $\psi\langle M\rangle = \psi_1\langle M\rangle \mathbf{U} (\psi_2\langle M\rangle \vee \mathbf{G}(\psi_1\langle\!\langle M\rangle\!\rangle))$ and $\psi\langle\!\langle M\rangle\!\rangle = \psi_1\langle\!\langle M\rangle\!\rangle \mathbf{W} \psi_2\langle\!\langle M\rangle\!\rangle$. The calculation above now holds after replacing the set $\{\psi\langle M\rangle, \psi\langle\!\langle M\rangle\!\rangle\}$ at the end of the right-hand side by $\{\psi\langle M\rangle, \psi\langle\!\langle M\rangle\!\rangle, \psi_2\langle M\rangle \vee \mathbf{G}(\psi_1\langle\!\langle M\rangle\!\rangle), \mathbf{G}(\psi_1\langle\!\langle M\rangle\!\rangle)\}$, so we obtain the bound $4|sf(\psi_1)| + 4|sf(\psi_2)| + 4 \leq 4|sf(\psi)|$. □

4.6 Evaluation

Our proof of the Normalization Theorem gives a good explanation of *why* the result holds. Roughly speaking, given a formula φ, the set of all words can be partitioned into contexts such that within each context the formula is equivalent to a formula of Σ_2 derived from φ by means of a very simple syntactic transformation. The procedure also provides a single exponential asymptotic upper bound for the length of the equivalent Δ_2-formula. However, the proof does not yield a normalization procedure efficient in practice. Indeed, a procedure based on Theorem 4.22 has exponential *best*-case complexity, since it requires to iterate over all subsets of B_{GF} and B_{FG}. Additionally, the procedure works top-down, and so it is particularly bad for families of formulas where, loosely speaking, the double alternation occurs near the bottom of the syntax tree.

Example 4.25. Consider the family of formulas

$$\varphi_n = (\cdots((((a_0 \mathbf{U} a_1) \mathbf{W} a_2) \mathbf{U} a_3) \mathbf{U} a_4) \cdots \mathbf{U} a_n)$$

for $n \geq 3$. The sets B_{GF} and B_{FG} have size $n - 1$ and 1, respectively. The normalization procedure based on the Normalization Theorem yields a disjunction of 2^{n+1} formulas, and so it takes exponential time in n. However, reasoning as at the beginning of Section 4.1, when we proved $\mathbf{FG}(a \mathbf{U} b) \equiv (\mathbf{GF}b \wedge \mathbf{FG}(a \mathbf{W} b))$, one can show that φ_n is equivalent to a Δ_2-formula of length $\Theta(n)$:

$$\begin{aligned} \varphi_n \quad \equiv \quad & (\mathbf{GF}a_1 \wedge (\cdots(((a_0 \mathbf{U} a_1) \mathbf{U} (a_2 \vee \mathbf{G}(a_0 \vee a_1))) \mathbf{U} a_3) \cdots \mathbf{U} a_n) \\ & \vee \ (\cdots(((a_0 \mathbf{U} a_1) \mathbf{U} a_2) \mathbf{U} a_3) \cdots \mathbf{U} a_n) \end{aligned}$$

Intuitively, to normalize φ_n, it suffices to solve the "local" problem caused by the subformula $((a_0 \mathbf{U} a_1) \mathbf{W} a_2) \mathbf{U} a_3$ of φ_n, which is in Σ_3; however, the procedure of Section 4 is blind to this fact and generates 2^{n+1} formulas.

5 A NORMALIZING REWRITE SYSTEM

We present a system of rewrite rules that allow us to normalize every LTL formula. As a corollary, we obtain an alternative proof of the Normalization Theorem. The normalization algorithms derived from the rewriting rules are more efficient than the ones presented in the previous section.

We first introduce a rewrite system for the fragment of the syntax without the operators $\mathbf{R}$ and $\mathbf{M}$. The reasons are purely expository. The restriction to this fragment makes the proofs shorter, and the extension to a rewrite system for the full syntax, presented in Section 5.6, is routine.

Remark 5.1. Recall also that the fragment without the operators $\mathbf{R}$ and $\mathbf{M}$ is as expressive as the full syntax. Indeed, exhaustively applying the well-known equivalences

$$\varphi_1\,\mathbf{R}\,\varphi_2 \equiv \varphi_1\,\mathbf{W}\,(\varphi_1 \wedge \varphi_2) \quad \text{and} \quad \varphi_1\,\mathbf{M}\,\varphi_2 \equiv \varphi_2\,\mathbf{U}\,(\varphi_1 \wedge \varphi_2)$$

to a formula yields an equivalent formula of the fragment. However, in the worst case, the formula of the fragment may be exponentially larger than the original one, and so eliminating the $\mathbf{R}$ and $\mathbf{M}$ operators and then applying the rewrite system for the fragment may be less efficient than a direct application of the rewrite system for the full syntax given in Section 5.6.

The key idea leading to the rewrite system is to treat the combinations $\mathbf{GF}$ (infinitely often) and $\mathbf{FG}$ (almost always) of temporal operators as *atomic* operators $\mathbf{G\!F}$ and $\mathbf{F\!G}$ (notice the typesetting with the two letters touching each other). We call them the *limit operators*; intuitively, whether a word satisfies a formula $\mathbf{G\!F}\varphi$ or $\mathbf{F\!G}\varphi$ depends only on its behavior "in the limit," in the sense that $w'w$ satisfies $\mathbf{G\!F}\varphi$ or $\mathbf{F\!G}\varphi$ iff w does.[2] So we add the limit operators to the fragment, yielding the following syntax.

Definition 5.2. Extended LTL formulas over a set Ap of atomic propositions are generated by the syntax:

$$\varphi ::= \mathbf{tt} \mid \mathbf{ff} \mid a \mid \neg a \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \varphi\,\mathbf{W}\,\varphi \mid \mathbf{G\!F}\varphi \mid \mathbf{F\!G}\varphi.$$

When determining the class of a formula in the syntactic future hierarchy, $\mathbf{G\!F}$ and $\mathbf{F\!G}$ are implicitly replaced by $\mathbf{GF}$ and $\mathbf{FG}$. For example, $\mathbf{F}\,\mathbf{G}\,\mathbf{F}a$ is rewritten into $\mathbf{F}\,\mathbf{G}\,\mathbf{F}a$, and so it is a formula of Σ_3.

> *Convention*: In the rest of the section we only consider extended formulas, which are by construction in negation normal form, and call them just *formulas*.

Let us now define our precise target normal form. Formulas of the form $\varphi\,\mathbf{U}\,\psi$, $\varphi\,\mathbf{W}\,\psi$, $\mathbf{X}\varphi$, $\mathbf{G\!F}\varphi$, and $\mathbf{F\!G}\varphi$ are called $\mathbf{U}$-, $\mathbf{W}$-, $\mathbf{X}$-, $\mathbf{G\!F}$-, and $\mathbf{F\!G}$-formulas, respectively.We refer to these formulas as *temporal* formulas. The syntax tree T_φ of a formula φ is defined in the usual way, and $|\varphi|$ denotes the number of nodes of T_φ. A node of T_φ is a $\mathbf{U}$-*node* if the subformula rooted at it is a $\mathbf{U}$-formula. $\mathbf{W}$-, $\mathbf{G\!F}$-, $\mathbf{F\!G}$- and temporal nodes are defined analogously.

Definition 5.3. Let φ be an LTL formula. A node of T_φ is a *limit node* if it is either a $\mathbf{G\!F}$-node or an $\mathbf{F\!G}$-node. The formula φ is in *normal form* if T_φ satisfies the following properties:

(1) No $\mathbf{U}$-node is under a $\mathbf{W}$-node.
(2) No limit node is under another temporal node.
(3) No $\mathbf{W}$-node is under a $\mathbf{G\!F}$-node, and no $\mathbf{U}$-node is under an $\mathbf{F\!G}$-node.

Remark 5.4. Observe that formulas in normal form belong to Δ_2. Even a slightly stronger statement holds: a formula in normal form is a positive Boolean combination of formulas of Σ_2 and formulas of the form $\mathbf{G\!F}\psi$ such that $\psi \in \Sigma_1$ (and so $\mathbf{G\!F}\psi \in \Pi_2$).

[2] Limit operators are called *suspendable* in the work of Babiak et al. [1].

There is a dual normal form in which property (1) is replaced by "no $\mathbf{W}$-node is under a $\mathbf{U}$-node," and the other two properties do not change. Formulas in dual normal form are positive Boolean combination of formulas of Π_2 and formulas of the form $\mathbf{FG}\psi$ such that $\psi \in \Pi_1$. Once the Normalization Theorem for the primal normal form is proved, a corresponding theorem for the dual form follows as an easy corollary (see Section 5.6).

We incrementally normalize formulas by dealing with the three requirements of the normal form, one by one, as follows:

(1) We remove all $\mathbf{U}$-nodes that are under some $\mathbf{W}$-node, but not under any limit node.
(2) We remove all limit nodes under some other temporal node.
(3) We remove all $\mathbf{W}$-nodes under some $\mathbf{GF}$-node, and all $\mathbf{U}$-nodes under some $\mathbf{FG}$-node.

The resulting formula is in normal form (Definition 5.3).

It is convenient to introduce intermediate normal forms corresponding to the results of applying the first and the first two steps. For this, let us introduce the following parameters of a formula φ:

- $n_u(\varphi)$ denotes the number of $\mathbf{U}$-nodes in T_φ that are under some $\mathbf{W}$-node, but not under any limit node of T_φ. For example, if $\varphi = (a\ \mathbf{U}\ b)\ \mathbf{W}\ (\mathbf{FG}(c\ \mathbf{U}\ d))$, then $n_u(\varphi) = 1$.
- $n_{\lim}(\varphi)$ denotes the number of distinct limit subformulas under some temporal operator. Formally, $n_{\lim}(\varphi)$ is the number of limit formulas ψ' such that ψ' is a proper subformula of a temporal subformula (proper or not) of φ. For example, if $\varphi = (\mathbf{FG}a\ \mathbf{U}\ \mathbf{GF}b) \vee (\mathbf{GF}b\ \mathbf{W}\ \mathbf{FG}a)$, then $n_{\lim}(\varphi) = 2$.

We define the normal forms as follows.

Definition 5.5. An LTL formula φ is in *1-form* if $n_u(\varphi) = 0$, and in *1-2-form* if $n_u(\varphi) = 0$ and $n_{\lim}(\varphi) = 0$.

Observe that after step (1), we obtain a formula in 1-form, and after steps (1) and (2), we obtain a formula in 1-2-form.

The rest of the section is structured as follows. Sections 5.1 through 5.3 present rewrite rules to conduct steps (1) through (3), respectively. Section 5.4 proves the Normalization Theorem and gives an upper bound in the size of the final formula. Section 5.5 summarizes the normalization procedure. Finally, Section 5.6 discusses some extensions and special cases.

5.1 Stage 1: Removing U-Nodes under W-Nodes

We consider formulas φ with placeholders—that is, "holes" that can be filled with a formula. Formally, let $[\quad]$ be a symbol denoting a special atomic proposition. A formula with placeholders is a formula with one or more occurrences of $[\quad]$, all of them positive (i.e., the formula has no occurrence of $\neg[\quad]$. We denote by $\varphi[\psi]$ the result of filling each placeholder of φ with an occurrence of ψ; formally, $\varphi[\psi]$ is the result of substituting ψ for $[\quad]$ in φ. For example, if $\varphi[\quad] = ([\quad]\mathbf{W}(a\mathbf{U}[\quad]))$, then $\varphi[\mathbf{X}b] = (\mathbf{X}b)\ \mathbf{W}\ (a\ \mathbf{U}\ \mathbf{X}b)$. We assume that $[\quad]$ binds more strongly than any operator (e.g., $\varphi_1\ \mathbf{W}\ \varphi_2[\psi] = \varphi_1\ \mathbf{W}\ (\varphi_2[\psi])$). We write $\varphi \equiv^w \psi$ as a shorthand for $w_k \models \varphi$ iff $w_k \models \psi$ for all $k \in \mathbb{N}$, the following lemma states two straightforward properties of formulas with placeholders.

Lemma 5.6. *For every formula with placeholder φ and every two formulas ψ and ψ',*

(1) If φ is in negative normal form (as every LTL formula considered in this article), $\psi \models \psi'$ implies $\varphi[\psi] \models \varphi[\psi']$.
(2) $\psi \equiv^w \psi'$ implies $\varphi[\psi] \equiv^w \varphi[\psi']$.

The following lemma allows us to pull $\mathbf{U}$-subformulas out of $\mathbf{W}$-formulas.

Lemma 5.7.

$$\varphi_1 \,\mathbf{W}\, \varphi_2[\psi_1 \,\mathbf{U}\, \psi_2] \;\equiv\; (\varphi_1 \,\mathbf{U}\, \varphi_2[\psi_1 \,\mathbf{U}\, \psi_2]) \vee \mathbf{G}\varphi_1 \tag{3}$$

$$\varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2 \;\equiv\; (\mathbf{GF}\psi_2 \wedge \varphi_1[\psi_1 \,\mathbf{W}\, \psi_2] \,\mathbf{W}\, \varphi_2) \vee \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}])) \tag{4}$$

Proof. For Equation (3), observe that, by the definition of the semantics of LTL, $\varphi_1 \,\mathbf{W}\, \varphi_2 \equiv \varphi_1 \,\mathbf{U}\, \varphi_2 \vee \mathbf{G}\varphi_2$ holds for arbitrary formulas φ_1, φ_2. For Equation (4), we proceed by invoking the Weak Contextual Equivalence Lemma (Lemma 4.3) with basis $B = \{\mathbf{GF}\psi_2\}$, main formula $\varphi = \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2$, and contextual formulas $\varphi\langle C \mid B\rangle = \varphi_1[\psi_1 \,\mathbf{W}\, \psi_2] \,\mathbf{W}\, \varphi_2$ for $C = \{\mathbf{GF}\psi_2\}$ and $\varphi\langle C \mid B\rangle = \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}]))$ for $C = \emptyset$. The conclusion of that lemma is exactly (4), but we have to prove its two premises.

First Premise: φ and $\varphi\langle C \mid B\rangle$ are equivalent under context $\langle C \mid B\rangle$.

For the case $C = \{\mathbf{GF}\psi_2\}$, let $w \models \mathbf{GF}\psi_2$. We have to show $\varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{W} \varphi_2 \equiv^w \varphi_1[\psi_1 \mathbf{W} \psi_2] \mathbf{W} \varphi_2$. By the semantics of LTL, we have $\psi_1 \mathbf{U} \psi_2 \equiv \psi_1 \mathbf{W} \psi_2 \wedge \mathbf{F}\psi_2$; further, $w \models \mathbf{GF}\psi_2$ implies $\mathbf{F}\psi_2 \equiv^w \mathbf{tt}$, and so we get $\psi_1 \mathbf{U} \psi_2 \equiv^w \psi_1 \mathbf{W} \psi_2$. Applying Lemma 5.6(2), we obtain $\varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{W} \varphi_2 \equiv^w \varphi_1[\psi_1 \mathbf{W} \psi_2] \mathbf{W} \varphi_2$, as desired.

For the case $C = \emptyset$, let $w \not\models \mathbf{GF}\psi_2$. We have to show $\varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2 \equiv^w \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}]))$. For $=\!\!|^w$, we prove the stronger $=\!\!|$:

$$\begin{aligned} &\varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}])) \\ &\quad \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\psi_1 \,\mathbf{U}\, \psi_2])) && \text{(Lemma 5.6(1) and } \mathbf{ff} \models \psi_1 \,\mathbf{U}\, \psi_2) \\ &\quad \equiv \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2 && \text{(semantics of LTL).} \end{aligned}$$

For $\models^w$, assume $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2$. We have to show $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}]))$. Consider two cases:

- $w_k \models \varphi_2$ for some $k \geq 0$. Then, since $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2$ by hypothesis, we have $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, \varphi_2$, and so $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}]))$.
- $w_k \not\models \varphi_2$ for every $k \geq 0$. Since $w \not\models \mathbf{GF}\psi_2$, there exists a smallest index n such that $w_k \not\models \psi_2$ for all $k \geq n$. Further, since $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{W}\, \varphi_2$ by hypothesis, we have in particular $w_k \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2]$ for every $k < n$. So to prove $w \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}]))$, it suffices to show $w_n \models \mathbf{G}\varphi_1[\mathbf{ff}]$, or, equivalently, that $w_k \models \varphi_1[\mathbf{ff}]$ holds for every $k \geq n$. For this, observe that $w_k \models \varphi_1[\psi_1 \,\mathbf{U}\, \psi_2]$ and $\psi_1 \,\mathbf{U}\, \psi_2 \equiv^{w_k} \mathbf{ff}$ hold for every $k \geq n$, the latter because $w_k \not\models \psi_2$ holds for every $k \geq n$. Apply now Lemma 5.6(2).

Second Premise: The formula for context $\emptyset$ entails the one for context $\{\mathbf{GF}\psi_2\}$.

$$\begin{aligned} &\varphi_1[\psi_1 \,\mathbf{U}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}])) \\ &\quad \models \varphi_1[\psi_1 \,\mathbf{W}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\mathbf{ff}])) && \text{(Lemma 5.6(1) and } \psi_1 \,\mathbf{U}\, \psi_2 \models \psi_1 \,\mathbf{W}\, \psi_2) \\ &\quad \models \varphi_1[\psi_1 \,\mathbf{W}\, \psi_2] \,\mathbf{U}\, (\varphi_2 \vee (\mathbf{G}\varphi_1[\psi_1 \,\mathbf{W}\, \psi_2])) && \text{(Lemma 5.6(1) and } \mathbf{ff} \models \psi_1 \,\mathbf{W}\, \psi_2) \\ &\quad \models \varphi_1[\psi_1 \,\mathbf{W}\, \psi_2] \,\mathbf{W}\, \varphi_2 && \text{(semantics of LTL)} \end{aligned}$$

□

Proposition 5.8. *For every LTL formula φ, there exists an equivalent formula φ' in 1-form such that $|\varphi'| \leq 4^{2|\varphi|} \cdot |\varphi|$. Moreover, for every subformula* $\mathbf{GF}\psi$ *of φ', the formula ψ is a subformula of φ, and every* $\mathbf{FG}$*-subformula of φ' is also a subformula of φ.*

Proof. We associate to each formula a rank, defined by $rank(\varphi) = |\varphi| + n_u(\varphi)$. Observe that a formula φ is in 1-form iff $rank(\varphi) = |\varphi|$. Throughout the proof, we say that a formula φ' *satisfies the limit property* if for every subformula $\mathbf{GF}\psi$ of φ' the formula ψ is a subformula of φ and every $\mathbf{FG}$-subformula of φ' is also a subformula of φ (notice the asymmetry). Further, we say that a formula φ' *satisfies the size property* if $|\varphi'| \leq 4^{rank(\varphi)} \cdot |\varphi|$ from which the claimed size bound immediately follows.

We prove by induction on $rank(\varphi)$ that φ is equivalent to a formula φ' in 1-form satisfying the limit and size properties. Within the inductive step, we proceed by a case distinction of φ:

If $\varphi = \mathbf{tt}, \mathbf{ff}, \mathbf{GF}\psi, \mathbf{FG}\psi$, then φ is already in 1-form and satisfies the limit and size properties.

If $\varphi = \varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2, \varphi_1 \mathbf{U} \varphi_2$, then by the induction hypothesis φ_1 and φ_2 can be normalized into formulas φ_1' and φ_2' satisfying the limit and size properties. The formulas $\varphi_1' \wedge \varphi_2', \varphi_1' \vee \varphi_2', \varphi_1' \mathbf{U} \varphi_2'$ are then in 1-form (the latter because the additional $\mathbf{U}$-node is above any $\mathbf{W}$-node) and satisfy the limit property. The size property holds because

$$\begin{aligned} |\varphi_1'| + |\varphi_2'| + 1 &\leq 4^{rank(\varphi_1)} \cdot |\varphi_1| + 4^{rank(\varphi_2)} \cdot |\varphi_2| + 1 && \text{(induction hypotheses)} \\ &\leq 4^{rank(\varphi_1)+rank(\varphi_2)} \cdot (|\varphi_1| + |\varphi_2| + 1) \\ &\leq 4^{rank(\varphi)} \cdot |\varphi| && (|\varphi_i| \leq |\varphi| \text{ and } \textstyle\sum_{i=1}^{2} rank(\varphi_i) < rank(\varphi)). \end{aligned}$$

If $\varphi = \mathbf{X}\varphi_1$, then by the induction hypothesis there is a formula φ_1' equivalent to φ_1 in 1-form, and so φ is equivalent to $\mathbf{X}\varphi_1'$, which is in 1-form and satisfies the limit and size properties.

If $\varphi = \varphi_1 \mathbf{W} \varphi_2$ and $n_u(\varphi) = 0$, then $\varphi_1 \mathbf{W} \varphi_2$ is already in 1-form and satisfies the limit and size properties.

If $\varphi = \varphi_1 \mathbf{W} \varphi_2$ and $n_u(\varphi) > 0$, then we proceed by a case distinction:

- φ_2 contains at least one $\mathbf{U}$-node that is not under a limit node. Let $\psi_1 \mathbf{U} \psi_2$ be such a $\mathbf{U}$-node. We derive $\varphi_2[\quad]$ from φ_2 by replacing each $\mathbf{U}$-node labeled by $\psi_1 \mathbf{U} \psi_2$ by the special atomic proposition $[\quad]$. By Lemma 5.7(3), we have

  $$\varphi_1 \mathbf{W} \varphi_2[\psi_1 \mathbf{U} \psi_2] \equiv \varphi_1 \mathbf{U} \varphi_2[\psi_1 \mathbf{U} \psi_2] \vee \varphi_1 \mathbf{W} \mathbf{ff}.$$

  Since $rank(\varphi_1) < rank(\varphi)$, $rank(\varphi_2) < rank(\varphi)$, and $rank(\varphi_1 \mathbf{W} \mathbf{ff}) < rank(\varphi)$ (the latter because φ_2 contains at least one $\mathbf{U}$-node), by the induction hypothesis φ_1, φ_2, and $\varphi_1 \mathbf{W} \mathbf{ff}$ can be normalized into formulas φ_1', φ_2', and φ_3' satisfying the limit and size properties. So φ can be normalized into $\varphi' = \varphi_1' \mathbf{U} \varphi_2' \vee \varphi_3'$. Moreover, φ' satisfies the limit property, because all $\mathbf{GF}$- and $\mathbf{FG}$-subformulas of φ' are subformulas of φ_1', φ_2', or φ_3'. For the size property, we calculate

$$\begin{aligned} |\varphi'| &= |\varphi_1'| + |\varphi_2'| + |\varphi_3'| + 2 \\ &\leq 4^{rank(\varphi_1)} \cdot |\varphi_1| + 4^{rank(\varphi_2)} \cdot |\varphi_2| + 4^{rank(\varphi_1 \mathbf{W} \mathbf{ff})} \cdot |\varphi_1 \mathbf{W} \mathbf{ff}| + 2 && \text{(induction hypotheses)} \\ &\leq 4^{rank(\varphi)-1} \cdot (|\varphi_1| + |\varphi_2| + |\varphi_1 \mathbf{W} \mathbf{ff}| + 2) \\ &\leq 4^{rank(\varphi)-1} \cdot 4 \cdot |\varphi| = 4^{rank(\varphi)} \cdot |\varphi|. \end{aligned}$$

- Every $\mathbf{U}$-node of φ_2 is under a limit node, and φ_1 contains at least one $\mathbf{U}$-node that is not under any limit node. Then φ_1 contains a maximal subformula $\psi_1 \mathbf{U} \psi_2$ (with respect to the subformula order) that is not under a limit node. We derive $\varphi_1[\quad]$ from φ_1 by replacing each $\mathbf{U}$-node labeled by $\psi_1 \mathbf{U} \psi_2$ that does not appear under a limit node by the special atomic proposition $[\quad]$. By Lemma 5.7(4), we have

  $$\varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{W} \varphi_2 \equiv \left(\mathbf{GF}\psi_2 \wedge \underbrace{\varphi_1[\psi_1 \mathbf{W} \psi_2] \mathbf{W} \varphi_2}_{\rho_1}\right) \vee \left(\underbrace{\varphi_1[\psi_1 \mathbf{U} \psi_2]}_{\rho_2} \mathbf{U} \underbrace{(\varphi_2 \vee (\varphi_1[\mathbf{ff}] \mathbf{W} \mathbf{ff})}_{\rho_3}\right).$$

  To apply the induction hypothesis, we argue that ρ_1, ρ_2, and ρ_3 have rank smaller than φ and thus can be normalized to ρ_1', ρ_2', and ρ_3' satisfying the limit and size properties. The formula ρ_1 has the same number of nodes as φ, but fewer $\mathbf{U}$-nodes under $\mathbf{W}$-nodes; so $n_u(\rho_1) < n_u(\varphi)$ and thus $rank(\rho_1) < rank(\varphi)$. The same argument applies to ρ_3. Finally, $rank(\rho_2) < rank(\varphi)$ follows from the fact that ρ_2 has fewer nodes than φ. So φ can be normalized to $\varphi' = (\mathbf{GF}\psi_2 \wedge \rho_1') \vee (\rho_2' \mathbf{U} \rho_3')$.

We show that φ' satisfies the limit property. Let $\mathbf{GF}\psi$ be a subformula of φ'. If $\mathbf{GF}\psi = \mathbf{GF}\psi_2$, then we are done, because ψ_2 is a subformula of φ. Otherwise, $\mathbf{GF}\psi$ is a subformula of ρ_1', ρ_2', or ρ_3'. Since all of them satisfy the limit property, ψ is a subformula of φ, and we are done. Further, every $\mathbf{FG}$-subformula of φ' belongs to ρ_1', ρ_2', or ρ_3' and so it is also subformula of φ. For the size property, we calculate

$$\begin{aligned}
|\varphi'| &= |\rho_1'| + |\rho_2'| + |\rho_3'| + |\psi_2| + 4 \\
&\leq 4^{rank(\rho_1)} \cdot |\rho_1| + 4^{rank(\rho_2)} \cdot |\rho_2| + 4^{rank(\rho_3)} \cdot |\rho_3| + |\varphi_1| + 4 && \text{(induction hypotheses)} \\
&\leq 4^{rank(\varphi)-1} \cdot (|\varphi| + |\varphi| + |\varphi|) + |\varphi| + 4 && (|\rho_i|, |\varphi_1| \leq |\varphi| \text{ and } rank(\rho_i) < rank(\varphi)). \\
&\leq 4^{rank(\varphi)-1} \cdot 4 \cdot |\varphi| = 4^{rank(\varphi)} \cdot |\varphi|
\end{aligned}$$

□

5.2 Stage 2: Moving GF- and FG-Subformulas Up

In this section, we address the second property of the normal form. The following lemma allows us to pull limit subformulas out of any temporal formula. (Note that the second rule is only necessary if the formula before stage 1 contained $\mathbf{FG}$-subformulas, since stage 1 only creates new $\mathbf{GF}$-formulas.)

Lemma 5.9.

$$\varphi[\mathbf{GF}\psi] \quad\equiv\quad (\mathbf{GF}\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}] \tag{5}$$

$$\varphi[\mathbf{FG}\psi] \quad\equiv\quad (\mathbf{FG}\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}] \tag{6}$$

Proof. We prove that $\varphi[\psi] \equiv (\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}]$ for every ψ such that $\mathbf{G}\psi \equiv \psi$—that is, $w \models \psi$ iff $w_k \models \psi$ for all $k \in \mathbb{N}$. This is a generalization of (5) and (6), since both $\mathbf{G}\,\mathbf{GF}\psi' \equiv \mathbf{GF}\psi'$ and $\mathbf{G}\,\mathbf{FG}\psi' \equiv \mathbf{FG}\psi'$ for any ψ'. The new statement follows from the Weak Contextual Equivalence Lemma (Lemma 4.3) with basis $\{\psi\}$ and formulas $\varphi[\psi]\langle\{\psi\}\rangle = \varphi[\mathbf{tt}]$ and $\varphi[\psi]\langle\emptyset\rangle = \varphi[\mathbf{ff}]$. It remains to prove that the premises of the lemma hold:

(i) Since ψ is satisfied by either all or no suffix of a word w, then either $\psi \equiv^w \mathbf{tt}$ or $\psi \equiv^w \mathbf{ff}$. Hence,

$$w \models \psi \overset{\mathbf{G}\psi\equiv\psi}{\iff} \psi \equiv^w \mathbf{tt} \overset{\text{Lemma 5.6(2)}}{\Rightarrow} \varphi[\psi] \equiv^w \varphi[\mathbf{tt}] \Rightarrow (w \models \varphi[\psi] \iff w \models \varphi[\mathbf{tt}]),$$

and identical arguments can be used to prove $w \models \varphi[\psi] \iff w \models \varphi[\mathbf{ff}]$ from $w \not\models \psi$.

(ii) Since $\mathbf{ff} \models \mathbf{tt}$ and by Lemma 5.6(1), $\varphi[\mathbf{ff}] \models \varphi[\mathbf{tt}]$.

□

We show using (5) and (6) that every formula in 1-form can be transformed into an equivalent formula in 1-2-form.

Proposition 5.10. *Every LTL formula φ in 1-form is equivalent to a formula φ' in 1-2-form such that $|\varphi'| \leq 3^{n_{lim}(\varphi)} \cdot |\varphi|$. Moreover, the size of the limit subformulas does not increase: for every $b > 0$, if $|\psi| \leq b$ for every limit subformula ψ of φ, then $|\psi'| \leq b$ for every limit subformula ψ' if φ'.*

Proof. We proceed by induction on the number of proper limit subformulas of φ. If φ does not contain any, then it is already in 1-2-form. Assume there exists such a proper limit subformula ψ that is smaller (or incomparable) to all other limit subformulas of φ according to the subformula order. We derive $\varphi[\quad]$ from φ by replacing each limit-node labeled by ψ by the special atomic proposition $[\quad]$. We then apply Lemma 5.9 to obtain

$$\varphi[\psi] \equiv (\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}], \qquad \text{where } \psi = \mathbf{GF}\psi', \mathbf{FG}\psi' .$$

Note that ψ does not properly contain any limit subformula, and so it is in 1-2-form. Both $\varphi[\mathbf{tt}]$ and $\varphi[\mathbf{ff}]$ are still in 1-form, and they have one limit operator less than φ. Thus, they can be normalized by the induction hypothesis into φ_1' and φ_2' in 1-2-form. Finally, $\varphi' = (\psi \wedge \varphi_1') \vee \varphi_2'$ is a Boolean combination of formulas in 1-2-form, so it is in 1-2-form. The number of nodes of $T_{\varphi'}$ can be crudely bounded as follows:

$$\begin{aligned}
|\varphi'| &\leq |\varphi_1'| + |\varphi_2'| + |\psi| + 2 \\
&\leq 3^{n_{\lim}(\varphi[\mathbf{tt}])} \cdot |\varphi[\mathbf{tt}]| + 3^{n_{\lim}(\varphi[\mathbf{ff}])} \cdot |\varphi[\mathbf{ff}]| + |\psi| + 2 && \text{(induction hypotheses)} \\
&\leq 3^{n_{\lim}(\varphi[\mathbf{tt}])} \cdot 2 \cdot |\varphi[\mathbf{tt}]| + |\psi| + 2 && (|\varphi[\mathbf{tt}]| = |\varphi[\mathbf{ff}]|, \text{idem for } n_{\lim}) \\
&\leq 3^{n_{\lim}(\varphi[\mathbf{tt}])} \cdot \big(2 \cdot (|\varphi| - |\psi| + 1) + |\psi| + 2\big) && (n_{\lim}(\varphi[\mathbf{tt}]) \leq n_{\lim}(\varphi) - 1) \\
&\leq 3^{n_{\lim}(\varphi)-1} \cdot (2|\varphi| - |\psi| + 4) \\
&\leq 3^{n_{\lim}(\varphi)-1} \cdot (3|\varphi|) = 3^{n_{\lim}(\varphi)} \cdot |\varphi| && (|\psi|, |\varphi| \geq 2).
\end{aligned}$$

To show that the size of the limit subformulas does not increase, let b be a bound on the size of the $\mathbf{GF}$-subformulas of φ. We claim that the size of each $\mathbf{GF}$-subformula of φ' is also bounded by b (the case of $\mathbf{FG}\psi$ is analogous). Indeed, the $\mathbf{GF}$-subformulas of φ' are ψ (which is already in φ) and the $\mathbf{GF}$-subformulas of φ_1' and φ_2'. Since the $\mathbf{GF}$-subformulas of $\varphi[\mathbf{tt}]$ and $\varphi[\mathbf{ff}]$ can only have decreased in size, by the induction hypothesis the number of nodes of any $\mathbf{GF}$-subformula of φ_1' and φ_2' is bounded by b, and we are done. □

5.3 Stage 3: Removing W-Nodes (U-Nodes) under GF-Nodes (FG-Nodes)

The normalization of LTL formulas is completed in this section by fixing the problems within limit subformulas. To do so, we introduce two new rewrite rules that allow us to pull $\mathbf{W}$-subformulas out of $\mathbf{GF}$-formulas, and $\mathbf{U}$-subformulas out of $\mathbf{FG}$-formulas.

Lemma 5.11.

$$\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2] \equiv \mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2] \vee (\mathbf{FG}\psi_1 \wedge \mathbf{GF}\varphi[\mathbf{tt}]) \tag{7}$$

$$\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2] \equiv (\mathbf{GF}\psi_2 \wedge \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]) \vee \mathbf{FG}\varphi[\mathbf{ff}] \tag{8}$$

Proof. Notice that (7) and (8) are instances of the Weak Contextual Equivalence Lemma (Lemma 4.3) with bases $\{\mathbf{FG}\psi_1\}$ and $\{\mathbf{GF}\psi_2\}$, respectively. In the case of (7), the contextual formulas are $(\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2])\langle\{\mathbf{FG}\psi_1\}\rangle = \mathbf{GF}\varphi[\mathbf{tt}]$ and $(\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2])\langle\emptyset\rangle = \mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2]$. The premises of the lemma are satisfied:

(i) Given $w \models \mathbf{FG}\psi_1$, we should first prove $w \models \mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$ iff $w \models \mathbf{GF}\varphi[\mathbf{tt}]$. Since $w \models \mathbf{FG}\psi_1$, there is a suffix v of w such that $v_k \models \psi_1$ for all $k \in \mathbb{N}$. By the semantics of LTL, $\psi_1 \mathbf{W} \psi_2 \equiv^v \mathbf{tt}$, so $\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2] \equiv^v \mathbf{GF}\varphi[\mathbf{tt}]$ by Lemma 5.6(2). Because $\mathbf{GF}$-formulas are satisfied in all suffixes of a word or in none of them, $w \models \mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$ iff $w \models \mathbf{GF}\varphi[\mathbf{tt}]$.
Given $w \not\models \mathbf{FG}\psi_1$, we claim $w \models \mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$ iff $w \models \mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2]$. We know $\psi_1 \mathbf{W} \psi_2 \equiv \psi_1 \mathbf{U} \psi_2 \vee \mathbf{G}\psi_1$, but $\mathbf{G}\psi_1 \equiv^w \mathbf{ff}$ because of the hypothesis, so $\psi_1 \mathbf{W} \psi_2 \equiv^w \psi_1 \mathbf{U} \psi_2$. By Lemma 5.6(2), $\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2] \equiv^w \mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2]$ and this implies the claim.

(ii) Since $\psi_1 \mathbf{U} \psi_2 \models \mathbf{tt}$ and by Lemma 5.6(1), $\mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2] \models \mathbf{GF}\varphi[\mathbf{tt}]$.

For (7), the formulas are $(\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2])\langle\{\mathbf{GF}\psi_2\}\rangle = \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]$ and $(\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2])\langle\emptyset\rangle = \mathbf{FG}\varphi[\mathbf{ff}]$, and the premises of the lemma also hold:

(i) Given $w \models \mathbf{GF}\psi_2$, we prove $w \models \mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2]$ iff $w \models \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]$. Since $\psi_1 \mathbf{U} \psi_2 \equiv \psi_1 \mathbf{W} \psi_2 \wedge \mathbf{F}\psi_2$ and $\mathbf{F}\psi_2 \equiv^w \mathbf{tt}$ by the hypothesis, $\psi_1 \mathbf{U} \psi_2 \equiv^w \psi_1 \mathbf{W} \psi_2$. By Lemma 5.6(2), $\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2] \equiv^w \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]$ and this entails the desired equivalence.

Given $w \not\models \mathbf{GF}\psi_2$, we have to prove $w \models \mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2]$ iff $w \models \mathbf{FG}\varphi[\mathbf{ff}]$. Since $w \not\models \mathbf{GF}\psi_2$, there is a suffix v of w such that $v_k \not\models \psi_2$ for all $k \in \mathbb{N}$. In this case, by the semantics of LTL, $\psi_1 \mathbf{U} \psi_2 \equiv^v \mathbf{ff}$, so $\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2] \equiv^v \mathbf{FG}\varphi[\mathbf{ff}]$ by Lemma 5.6(2). The conclusion follows again from the limit properties of $\mathbf{FG}$.

(ii) Since $\mathbf{ff} \models \psi_1 \mathbf{W} \psi_2$ and by Lemma 5.6(1), $\mathbf{FG}\varphi[\mathbf{ff}] \models \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]$. □

The following proposition repeatedly applies these rules to show that limit formulas can be normalized with an exponential blowup.

Proposition 5.12. *For every LTL formula φ without limit operators, $\mathbf{GF}\varphi$ and $\mathbf{FG}\varphi$ can be normalized into formulas with at most $|\varphi'| \leq 3^{|\varphi|} \cdot |\varphi|$ nodes.*

Proof. A $\mathbf{GF}$-obstacle of a formula is a $\mathbf{W}$-node or a $\mathbf{U}$-node under a $\mathbf{W}$-node inside a $\mathbf{GF}$-node. Similarly, an $\mathbf{FG}$-obstacle is a $\mathbf{U}$-node or a $\mathbf{W}$-node under a $\mathbf{U}$-node inside a $\mathbf{FG}$-node. Finally, an obstacle is either a $\mathbf{GF}$-obstacle or an $\mathbf{FG}$-obstacle. We proceed by induction on the number of obstacles of $\mathbf{GF}\varphi$ or $\mathbf{FG}\varphi$. If they have no obstacles, then they are already in normal form (Definition 5.3).

Assume $\mathbf{GF}\varphi$ has at least one obstacle. Then φ contains at least one maximal $\mathbf{W}$-node $\psi_1 \mathbf{W} \psi_2$. We derive $\mathbf{GF}\varphi[\quad]$ from $\mathbf{GF}\varphi$ by replacing each $\mathbf{W}$-node labeled by $\psi_1 \mathbf{W} \psi_2$ by the special atomic proposition $[\quad]$. By Lemma 5.11, $\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$ is equivalent to

$$\mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2] \vee (\mathbf{FG}\psi_1 \wedge \mathbf{GF}\varphi[\mathbf{tt}]).$$

We claim that each of $\mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2]$, $\mathbf{GF}\varphi[\mathbf{tt}]$, and $\mathbf{FG}\psi_1$ has fewer obstacles than $\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$ and so can be normalized by the induction hypothesis. Indeed, $\mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2]$, and $\mathbf{GF}\varphi[\mathbf{tt}]$ have at least one $\mathbf{W}$-node less than φ, and the number of $\mathbf{U}$-nodes under a $\mathbf{W}$-node, due to the maximality of $\psi_1 \mathbf{W} \psi_2$, has not increased, and as a consequence it has fewer $\mathbf{GF}$-obstacles (and by definition no $\mathbf{FG}$-obstacles). For $\mathbf{FG}\psi_1$, observe first that every $\mathbf{FG}$-obstacle of $\mathbf{FG}\psi_1$ is a $\mathbf{GF}$-obstacle of $\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$. Indeed, the obstacles of $\mathbf{FG}\psi_1$ are the $\mathbf{U}$-nodes and the $\mathbf{W}$-nodes under $\mathbf{U}$-nodes; the former were $\mathbf{U}$-nodes under $\mathbf{W}$-nodes in φ, and the latter were $\mathbf{W}$-nodes of φ, and so both $\mathbf{GF}$-obstacles of $\mathbf{GF}\varphi$. Moreover, $\psi_1 \mathbf{W} \psi_2$ is a $\mathbf{GF}$-obstacle of φ, but not an $\mathbf{FG}$-obstacle of $\mathbf{FG}\psi_1$. Hence, the number of obstacles has decreased.

Assume now that $\mathbf{FG}\varphi$ has at least one obstacle. Then φ contains at least one maximal $\mathbf{U}$-node $\psi_1 \mathbf{U} \psi_2$. We derive $\mathbf{FG}\varphi[\quad]$ from $\mathbf{FG}\varphi$ by replacing each $\mathbf{U}$-node labeled by $\psi_1 \mathbf{U} \psi_2$ by the special atomic proposition $[\quad]$. By Lemma 5.11, $\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2]$ is equivalent to

$$(\mathbf{GF}\psi_2 \wedge \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]) \vee \mathbf{FG}\varphi[\mathbf{ff}].$$

Each of $\mathbf{GF}\psi_2$, $\mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]$, and $\mathbf{FG}\varphi[\mathbf{ff}]$ has fewer obstacles as $\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2]$ and can be normalized by the induction hypothesis. The proof is as above.

The size of the formula increases at most by a factor of 3 on each step, and the number of steps is bounded by the number of both $\mathbf{W}$-nodes and $\mathbf{U}$-nodes in φ, which is bounded by the total number of nodes in φ. So the formula has at most $3^{|\varphi|}|\varphi|$ nodes. □

5.4 The Normalization Theorem

The main result directly follows from the previous propositions.

Theorem 5.13. *Every formula φ of LTL is normalizable into a formula with at most $4^{6|\varphi|}$ nodes.*

Proof. Any LTL formula φ can be transformed into an equivalent φ' in 1-form of size $|\varphi'| \leq 4^{2|\varphi|} \cdot |\varphi|$ by Proposition 5.8. Moreover, $n_{\text{lim}}(\varphi') \leq 2 \cdot |\varphi|$, since every $\mathbf{FG}$-subformula of ψ' and every argument ψ of a $\mathbf{GF}$-subformula of φ' is a subformula of φ. In addition, $|\psi| \leq |\varphi|$ for every $\mathbf{GF}\psi$ subformula of φ.

Table 1. Normalization Rules

Stage 1:	$\varphi_1 \mathbf{W} \varphi_2[\psi_1 \mathbf{U} \psi_2]$	$\equiv$	$\varphi_1 \mathbf{U} \varphi_2[\psi_1 \mathbf{U} \psi_2] \vee \mathbf{G}\varphi_1$	(3)
	$\varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{W} \varphi_2$	$\equiv$	$(\mathbf{GF}\psi_2 \wedge \varphi_1[\psi_1 \mathbf{W} \psi_2] \mathbf{W} \varphi_2) \vee \varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{U} (\varphi_2 \vee \mathbf{G}\varphi_1[\mathbf{ff}])$	(4)
Stage 2:	$\varphi[\mathbf{GF}\psi]$	$\equiv$	$(\mathbf{GF}\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}]$	(5)
	$\varphi[\mathbf{FG}\psi]$	$\equiv$	$(\mathbf{FG}\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}]$	(6)
Stage 3:	$\mathbf{GF}\varphi[\psi_1 \mathbf{W} \psi_2]$	$\equiv$	$\mathbf{GF}\varphi[\psi_1 \mathbf{U} \psi_2] \vee (\mathbf{FG}\psi_1 \wedge \mathbf{GF}\varphi[\mathbf{tt}])$	(7)
	$\mathbf{FG}\varphi[\psi_1 \mathbf{U} \psi_2]$	$\equiv$	$(\mathbf{GF}\psi_2 \wedge \mathbf{FG}\varphi[\psi_1 \mathbf{W} \psi_2]) \vee \mathbf{FG}\varphi[\mathbf{ff}]$	(8)

According to Proposition 5.10, for every formula φ' in 1-form, there is an equivalent formula φ'' in 1-2-form with

$$|\varphi''| \leq 3^{n_{\lim}(\varphi')} \cdot |\varphi'| \leq 3^{2|\varphi|} \cdot (4^{2|\varphi|} \cdot |\varphi|) \leq 3^{2|\varphi|} \cdot 4^{3|\varphi|} \tag{9}$$

This formula is a Boolean combination of limit formulas with at most $|\varphi|$ nodes, not containing any proper limit node, and other temporal formulas containing neither limit nodes nor **U**-nodes under **W**-nodes. The latter are in Σ_2, and Proposition 5.12 deals with the former. Notice that every $\mathbf{GF}\psi$ and $\mathbf{FG}\psi$ subformula has at most $|\varphi|$ nodes and thus can be normalized into a formula with at most $3^{|\varphi|}|\varphi|$ nodes. The result φ''' of replacing these limit subformulas by their normal forms within φ'' is a Boolean combination of normal forms, and so we are done. The number of nodes in the resulting formula φ''' is at most

$$\begin{aligned}
|\varphi'''| &\leq |\varphi''| + n_{\lim}(\varphi'') \cdot 3^{|\varphi|} \cdot |\varphi| \\
&\leq |\varphi''| \cdot 3^{|\varphi|} \cdot (|\varphi| + 1) && (n_{\lim}(\varphi'') \leq |\varphi''|) \\
&\leq 4^{3|\varphi|} \cdot 3^{3|\varphi|} \cdot (|\varphi| + 1) && (\text{by } (9)) \\
&\leq 4^{3|\varphi|} \cdot 4^{(3\log_4 3 + \frac{1}{2})|\varphi|} \leq 4^{6|\varphi|} && (|\varphi| + 1 \leq 4^{|\varphi|/2}).
\end{aligned}$$

□

5.5 Summary of the Normalization Algorithm

We summarize the steps of the normalization algorithm described and proven in this section. Recall that a formula is in normal form iff it satisfies the following properties:

(1) No **U**-node is under a **W**-node.
(2) No limit node is under another temporal node.
(3) No **W**-node is under a **GF**-node, and no **U**-node is under an **FG**-node.

The normalization algorithm applies the rules in Table 1 as follows to fix any violation of these properties:

(1) **U**-nodes under **W**-nodes and not under limit nodes are removed using rules (3) and (4). This may introduce new **GF**-subformulas. By applying (4) only to highest **U**-nodes of φ_1, the number of new **GF**-subformulas is only linear in the size of the original formula.
(2) Limit nodes under other temporal nodes are pulled out using rules (5) and (6). By applying the rules only to the lowest limit nodes, it only needs to be applied once for each limit subformula.

Table 2. Normalization Rules for $\mathbf{R}$ and $\mathbf{M}$

Stage 1:	$\varphi_1[\psi_1 \mathbf{M} \psi_2] \mathbf{W} \varphi_2$	$\equiv$	$(\mathbf{GF}\psi_1 \wedge \varphi_1[\psi_1 \mathbf{R} \psi_2] \mathbf{W} \varphi_2) \vee \varphi_1[\psi_1 \mathbf{M} \psi_2] \mathbf{U} (\varphi_2 \vee \mathbf{G}\varphi_1[\mathbf{ff}])$
	$\varphi_1 \mathbf{W} \varphi_2[\psi_1 \mathbf{M} \psi_2]$	$\equiv$	$\varphi_1 \mathbf{U} \varphi_2[\psi_1 \mathbf{M} \psi_2] \vee \mathbf{G}\varphi_1$
	$\varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{R} \varphi_2$	$\equiv$	$\varphi_1[\psi_1 \mathbf{U} \psi_2] \mathbf{M} \varphi_2 \vee \mathbf{G}\varphi_2$
	$\varphi_1[\psi_1 \mathbf{M} \psi_2] \mathbf{R} \varphi_2$	$\equiv$	$\varphi_1[\psi_1 \mathbf{M} \psi_2] \mathbf{M} \varphi_2 \vee \mathbf{G}\varphi_2$
	$\varphi_1 \mathbf{R} \varphi_2[\psi_1 \mathbf{U} \psi_2]$	$\equiv$	$(\mathbf{GF}\psi_2 \wedge \varphi_1 \mathbf{R} \varphi_2[\psi_1 \mathbf{W} \psi_2]) \vee (\varphi_1 \vee \mathbf{G}\varphi_2[\mathbf{ff}]) \mathbf{M} \varphi_2[\psi_1 \mathbf{U} \psi_2]$
	$\varphi_1 \mathbf{R} \varphi_2[\psi_1 \mathbf{M} \psi_2]$	$\equiv$	$(\mathbf{GF}\psi_1 \wedge \varphi_1 \mathbf{R} \varphi_2[\psi_1 \mathbf{R} \psi_2]) \vee (\varphi_1 \vee \mathbf{G}\varphi_2[\mathbf{ff}]) \mathbf{M} \varphi_2[\psi_1 \mathbf{M} \psi_2]$
Stage 3:	$\mathbf{GF}\varphi[\psi_1 \mathbf{R} \psi_2]$	$\equiv$	$\mathbf{GF}\varphi[\psi_1 \mathbf{M} \psi_2] \vee (\mathbf{FG}\psi_2 \wedge \mathbf{GF}\varphi[\mathbf{tt}])$
	$\mathbf{FG}\varphi[\psi_1 \mathbf{M} \psi_2]$	$\equiv$	$(\mathbf{GF}\psi_1 \wedge \mathbf{FG}\varphi[\psi_1 \mathbf{R} \psi_2]) \vee \mathbf{FG}\varphi[\mathbf{ff}]$

(3) $\mathbf{W}$-nodes under $\mathbf{GF}$-nodes are removed using rule (7), and $\mathbf{U}$-nodes under $\mathbf{FG}$-nodes are removed using rule (8). This may produce new limit nodes of smaller size that are handled recursively. Choosing the highest $\mathbf{W}$- and $\mathbf{U}$-nodes ensures that the process produces only a single exponential blowup over the initial size of the formula.

After the three steps, a formula in normal form is obtained with a single exponential blowup in the number of nodes.

Moreover, notice that ψ_1 itself does not play any role in rules (3) and (8), and neither does ψ_2 in (7). Hence, the application of (3) can be made more efficient by replacing not only every occurrence of $\psi_1 \mathbf{U} \psi_2$ outside a limit subformula with $\psi_1 \mathbf{W} \psi_2$ and $\mathbf{ff}$, but also every occurrence of $\psi \mathbf{U} \psi_2$ for any formula ψ by $\psi \mathbf{W} \psi_2$ and by $\mathbf{ff}$. The same holds for rules (7) and (8).

Example 5.14. Let us apply the procedure to the formula $\varphi_n = (\cdots((((a_0 \mathbf{U} a_1) \mathbf{W} a_2) \mathbf{U} a_3) \mathbf{U} a_4) \cdots \mathbf{U} a_n)$ in Example 4.25. In stage 1, rule (4) matches the subformula $(a_0 \mathbf{U} a_1) \mathbf{W} a_2$ and rewrites it to $\mathbf{GF}a_1 \wedge (a_0 \mathbf{W} a_1) \mathbf{W} a_2 \vee (a_0 \mathbf{U} a_1) \mathbf{U} (a_2 \vee \mathbf{ff} \mathbf{W} \mathbf{ff})$, where $\mathbf{ff} \mathbf{W} \mathbf{ff}$ can be simplified to $\mathbf{ff}$ and removed from the disjunction. The rewritten formula is in 1-form, because there is no $\mathbf{U}$-node under a $\mathbf{W}$-node, so we can continue to stage 2. Now, we must pull the $\mathbf{GF}$-node $\mathbf{GF}a_1$ out the cascade of $\mathbf{U}$-nodes using rule (5). This yields

$$\begin{aligned} \varphi_n \quad \equiv \quad & (\mathbf{GF}a_1 \wedge (\cdots((((a_0 \mathbf{W} a_1) \mathbf{W} a_2 \vee (a_0 \mathbf{U} a_1) \mathbf{U} a_2) \mathbf{U} a_3) \mathbf{U} a_4) \cdots \mathbf{U} a_n)) \\ & \vee (\cdots((((a_0 \mathbf{U} a_1) \mathbf{U} a_2) \mathbf{U} a_3) \mathbf{U} a_4) \cdots \mathbf{U} a_n). \end{aligned}$$

Since the only remaining limit node is outside any temporal formula, we have obtained a formula in 1-2-form and the procedure arrives to stage 3. Again, the only limit subformula is $\mathbf{GF}a_1$, and a_1 does not contain any $\mathbf{W}$-node, so the formula is completely normalized and we have finished. Observe that φ_n has been normalized by exactly two rule applications for all $n \geq 3$, so the algorithm proceeds in linear time for this family of formulas. The result is not identical, but very similar to the one in Example 4.25.

5.6 Extensions and Fragments

The Operators $\mathbf{R}$ *and* $\mathbf{M}$. We have omitted these operators from the proof and the normalization procedure, since they can be expressed in terms of the subset of operators we have considered. However, this translation exponentially increases the number of nodes of the formula, so handling

Table 3. LTL[**F**, **G**, **X**]-Normalization Rules

Stage 1:	$\mathbf{G}(\varphi) \equiv$	$\bigwedge_{C \in \mathsf{cnf}(\varphi)} \mathbf{G}(\bigvee_{\psi \in C} \psi)$ if φ is not a disjunction of temporal nodes	(3*)
	$\mathbf{G}(\mathbf{F}\psi \vee \varphi[\mathbf{F}\psi]) \equiv$	$\mathbf{GF}\psi \vee \mathbf{F}(\psi \wedge \mathbf{XG}\varphi[\mathbf{ff}]) \vee \mathbf{G}(\varphi[\mathbf{ff}])$	(4*)
Stage 2:	$\varphi[\mathbf{GF}\psi] \equiv$	$(\mathbf{GF}\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}]$	(5)
	$\varphi[\mathbf{FG}\psi] \equiv$	$(\mathbf{FG}\psi \wedge \varphi[\mathbf{tt}]) \vee \varphi[\mathbf{ff}]$	(6)
Stage 3:	$\mathbf{GF}\varphi[\mathbf{G}\psi] \equiv$	$\mathbf{GF}\varphi[\mathbf{ff}] \vee (\mathbf{FG}\psi \wedge \mathbf{GF}\varphi[\mathbf{tt}])$	(7*)
	$\mathbf{FG}\varphi[\mathbf{F}\psi] \equiv$	$(\mathbf{GF}\psi \wedge \mathbf{FG}\varphi[\mathbf{tt}]) \vee \mathbf{FG}\varphi[\mathbf{ff}]$	(8*)

them directly is convenient for efficiency. Their role at every step of the procedure is analogous to that of the **U** and **W** operators—that is, we treat **R** in the same way as **W**, and we treat **M** in the same way as **U**. The corresponding rules are shown in Table 2.

Dual Normal Form. Recall that a formula is in dual normal form if it satisfies conditions 2 and 3 of Definition 5.3 and no **W**-node is under a **U**-node. Given a formula φ, let ψ be a formula in primal normal form equivalent to $\overline{\varphi}$. Since $\varphi \equiv \neg\overline{\varphi} \equiv \neg\psi$, pushing the negation into ψ yields a formula equivalent to φ in dual normal form.

Past LTL. Past LTL is an extension of LTL with past operators like yesterday (**Y**), since (**S**), and so forth. In an appendix in the work of Gabbay [15], eight rewrite rules to pull future operators out of past operators are introduced. Combining these rules with ours yields a procedure that transforms a Past LTL formula into a normalized LTL formula, where past operators are gathered in past-only subformulas, and so can be considered atomic propositions.

LTL[**F**, **G**, **X**]. Note that LTL is equivalent to first-order logic (FO) over words [10]. In particular, a translation from LTL to FO that uses only three variables can be obtained by replacing LTL operators by their respective semantic definitions which are already FO-formulas (see Definition 2.2.). Thus, in fact, LTL is equivalent to FO[3], where 3 denotes the number of variables that are used. If we restrict LTL to the unary temporal operators (**F**, **G**, and **X**) and apply the same idea, we obtain formulas with at most two variables (FO[2]). Indeed, FO[2] is equivalent to LTL[**F**, **G**, **X**] [14]. However, our normalization procedure does not take this into account: if we start with a formula from LTL[**F**, **G**, **X**] and apply Theorem 5.13, we do not necessarily obtain a normalized formula from LTL[**F**, **G**, **X**]. Is the introduction of **U** that are not equivalent to **F** unavoidable?

Luckily, no. We can update our rewrite rules such that if we start with a formula from LTL[**F**, **G**, **X**], we also obtain a normalized formula from LTL[**F**, **G**, **X**]. The rules are in Table 3, and the correctness proofs proceed as before. The idea is now that we proceed by a case distinction over each $\mathbf{F}\psi$ and split into three cases: (a) $\mathbf{F}\psi$ never holds; (b) $\mathbf{F}\psi$ holds at least once, but finitely often; and (c) $\mathbf{F}\psi$ always holds. Moreover, since $\mathbf{XF}\psi \equiv \mathbf{FX}\psi$ and $\mathbf{XG}\psi \equiv \mathbf{GX}\psi$, every **G**-subformula with a nested **F**-subformula can always be reduced to one or more formulas $\mathbf{G}(\mathbf{F}\psi \vee \mathbf{G}\varphi[\mathbf{F}\psi])$ by pushing next operators inside **F** and **G**, calculating the conjunctive normal form of the argument, and splitting the **G** operator over the conjunctions.

6 A NOVEL TRANSLATION FROM LTL TO DETERMINISTIC RABIN AUTOMATA (DRW)

We apply our Δ_2-normalization procedure to derive a new translation from LTL to DRW via weak alternating automata (AWW).

It is well known that an LTL formula φ of length n can be translated into an AWW with $O(n)$ states [36, 47]. We show that if φ is in normal form, then the AWW can be chosen so that every path through the automaton switches at most once between accepting and non-accepting states. We then prove that determinizing AWWs satisfying this additional property is much simpler than the general case. Finally, we give an upper bound on the size of the resulting DRW: if we use the normal form of Theorem 4.22, then the number of states of the final DRW is double exponential in the length of φ, which is asymptotically optimal.

The section is structured as follows. Section 6.1 introduces basic definitions, Section 6.2 shows how to translate an Δ_2-formula into AWWs with at most one alternation between accepting and non-accepting states, and Section 6.3 presents the determinization procedure for this subclass of AWWs.

6.1 Weak and Very Weak Alternating Automata

Let X be a finite set. The set of positive Boolean formulas over X, denoted by $\mathcal{B}^+(X)$, is the closure of $X \cup \{\mathbf{tt}, \mathbf{ff}\}$ under disjunction and conjunction. A set $S \subseteq X$ is a model of $\theta \in B^+(X)$ if the truth assignment that assigns true to the elements of S and false to the elements of $X \setminus S$ satisfies θ. Observe that if S is a model of θ and $S \subseteq S'$, then S' is also a model of θ. A model S of θ is minimal if no proper subset of S is also a model of θ. The set of minimal models of θ is denoted $\mathcal{M}_\theta$. Observe that $\mathcal{M}_{\mathbf{tt}} = \{\emptyset\}$ and $\mathcal{M}_{\mathbf{ff}} = \emptyset$. Two formulas are equivalent, denoted $\theta \equiv \theta'$, if their sets of minimal models are equal (i.e., $\mathcal{M}_\theta = \mathcal{M}_{\theta'}$).

Alternating Büchi and Co-Büchi Automata. An alternating Büchi automaton over an alphabet Σ is a tuple $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$, where Q is a finite set of states, $\theta_0 \in \mathcal{B}^+(Q)$ is an initial formula,[3] $\delta\colon Q \times \Sigma \mapsto \mathcal{B}^+(Q)$ is the transition function, and $\alpha \subseteq Q$ is the acceptance condition [7, 48].

Example 6.1. We use as running example the alternating Büchi automaton $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$ over the alphabet $\Sigma = 2^{\{a,b,c\}}$, where $Q = \{q_0, q_1, q_2\}$, $\theta_0 = q_0$, $\alpha = \{q_1\}$, and δ is given by

$$\delta(q_0, \sigma) = \begin{cases} q_0 \vee q_1 & \text{if } \sigma = \{a\} \\ q_2 & \text{if } \sigma = \{b\} \\ \mathbf{ff} & \text{otherwise,} \end{cases} \quad \delta(q_1, \sigma) = \begin{cases} q_1 & \text{if } \sigma = \{b\} \\ q_1 \wedge q_2 & \text{otherwise,} \end{cases} \quad \text{and} \quad \delta(q_2, \sigma) = \begin{cases} \mathbf{tt} & \text{if } \sigma = \{c\} \\ q_2 & \text{otherwise.} \end{cases}$$

A *run* of $\mathcal{A}$ on the word w is a directed acyclic graph $G = (V, E)$ satisfying the following properties:

- $V \subseteq Q \times \mathbb{N}_0$, and $E \subseteq \{((q,l),(q',l+1)) \mid q, q' \in Q \text{ and } l \geq 0\}$. We call the elements of V *nodes.* For every $l \geq 0$, we let $Q_l := \{q\colon (q,l) \in V\}$.
- Q_0 is a minimal model of θ_0.
- For every $l \geq 0$ and every $q \in Q_l$, either $\delta(q, w[l]) \equiv \mathbf{ff}$ or the set $\{q'\colon ((q,l),(q',l+1)) \in E\}$ is a minimal model of $\delta(q, w[l])$. (Observe that if $\delta(q, w[l]) \equiv \mathbf{tt}$, then this set is empty.)
- For every $l \geq 1$, if $q \in Q_l$, then there exists $q' \in Q_{l-1}$ such that $((q', l-1),(q,l)) \in E$.

Runs can be finite or infinite. A run G is *accepting* if

(a) $\delta(q, w[l]) \not\equiv \mathbf{ff}$ for every $q \in Q_l$, and
(b) every infinite path of G visits infinitely often nodes $(q,l) \in V$ such that $q \in \alpha$.

In particular, every finite run satisfying (a) is accepting. $\mathcal{A}$ accepts a word w iff it has an accepting run G on w. The language $\mathcal{L}(\mathcal{A})$ recognized by $\mathcal{A}$ is the set of words accepted by $\mathcal{A}$. Two alternating Büchi automata are equivalent if they recognize the same language.

[3] Many papers define alternating automata with an initial state instead of an initial formula. Both models are equivalent, and an initial formula is better for our purposes.

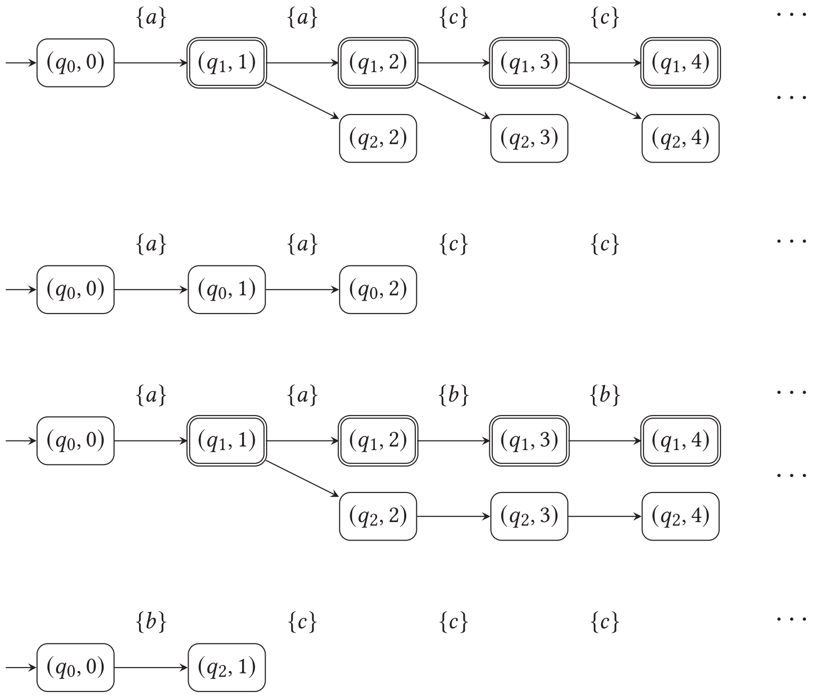


Fig. 2. Initial parts of four runs of the alternating Büchi automaton of Example 6.1 on the words $\{a\}\{a\}\{c\}^\omega$, $\{a\}\{a\}\{b\}^\omega$, and $\{b\}\{c\}^\omega$.

Example 6.2. Figure 2 shows (initial parts of) four runs of the alternating Büchi automaton of Example 6.1 on the words $\{a\}\{a\}\{c\}^\omega$ (top two runs), $\{a\}\{a\}\{b\}^\omega$ (third run), and $\{b\}\{c\}^\omega$ (fourth run). Let us check that the first run satisfies the properties of the definition. We have $Q_0 = \{q_0\}$, which is indeed a minimal model of θ_0. Further, we have the following:

- The set $\{q' \colon ((q_0, 0), (q', 1)) \in E\} = \{q_1\}$ is a minimal model of $\delta(q_0, \{a\}) = q_0 \vee q_1$.
- For every $l \geq 1$, the set $\{q' \colon ((q_1, l), (q', l+1)) \in E\} = \{q_1, q_2\}$ is a minimal model of $\delta(q_1, \{a\}) = \delta(q_1, \{c\}) = q_1 \wedge q_2$.
- For every $l \geq 2$, the set $\{q' \colon ((q_2, l, (q', l+1)) \in E\} = \emptyset$ is a minimal model of $\delta(q_2, \{c\}) = \mathbf{tt}$.

The first run is infinite and accepting, because the only infinite path of the run visits q_1 infinitely often. The second run is finite and not accepting, because $\delta(q_0, \{c\}) = \mathbf{ff}$. The third run is infinite and also not accepting, because the infinite path $(q_0, 0)(q_1, 1)(q_2, 2)(q_2, 3)(q_2, 4) \cdots$ only visits one node with states of α, namely $(q_1, 1)$. The fourth run is finite and accepting, because $\delta(q_2, \{c\}) = \mathbf{tt}$.

Alternating *co-Büchi* automata are defined as Büchi automata, replacing condition (b) by the co-Büchi condition (every infinite path of G only visits α-nodes finitely often).

An alternating automaton is *deterministic* if for every state $q \in Q$ and every letter $a \in \Sigma$ there exists $q' \in Q$ such that $\delta(q, a) = q'$, and *non-deterministic* if for every $q \in Q$ and every $a \in \Sigma$ there exists $Q' \subseteq Q$ such that $\delta(q, a) = \bigvee_{q' \in Q'} q'$.

Weak and Very Weak Automata. Let $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$ be an alternating (co-)Büchi automaton. We write $q \longrightarrow q'$ if there is $a \in \Sigma$ such that q' belongs to some minimal model of $\delta(q, a)$. An automaton $\mathcal{A}$ is *weak* if there is a partition $Q_0, \ldots, Q_m$ of Q such that

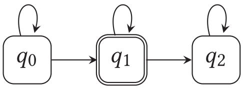


Fig. 3. Relation $\longrightarrow$ between the states of the automaton of Example 6.1.

- for every $q, q' \in Q$, if $q \longrightarrow q'$, then there are $i \leq j$ such that $q \in Q_i$ and $q' \in Q_j$, and
- for every $0 \leq i \leq m$, $Q_i \subseteq \alpha$ or $Q_i \cap \alpha = \emptyset$.

$\mathcal{A}$ is *very weak* or *linear* if it is weak and every class Q_i of the partition is a singleton (i.e., $|Q_i| = 1$) [23, 35]. We let AWW and A1W denote the set of weak and very weak alternating automata, respectively.

Example 6.3. Figure 3 shows the relation $\longrightarrow$ between the states of the automaton of Example 6.1. For example, we have $q_1 \longrightarrow q_2$ because $\{q_1, q_2\}$ is a minimal model of $\delta(q_1, \{a\}) = q_1 \wedge q_2$. The automaton is very weak because of the partition $Q_0 = \{q_0\}$, $Q_1 = \{q_1\}$, $Q_2 = \{q_2\}$.

Observe that for every *weak* automaton with a co-Büchi acceptance condition, we can define a Büchi acceptance condition on the same structure recognizing the same language. Thus, we will from now on assume that every weak automaton is equipped with a Büchi acceptance condition.

We define the alternation height of a weak alternating automaton. The definition is very similar, but not identical, to the standard one as presented, for example, in the work of Vardi [48]. A weak automaton $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$ has *alternation height n* if every path $q \rightarrow q' \rightarrow q'' \cdots$ of $\mathcal{A}$ alternates at most $n - 1$ times between α and $Q \setminus \alpha$. The automaton of Example 6.1 (see also Figure 3) has height 3 because of the path $q_0 \rightarrow q_1 \rightarrow q_2$, which exhibits two alternations.

Definition 6.4. AWW$[n]$ (respectively, A1W$[n]$) denote the set of all weak (respectively very weak) alternating automata with height at most n. Further, we let AWW$[n, \mathbf{A}]$ (respectively, AWW$[n, \mathbf{R}]$) denote the set of automata of AWW$[n]$ whose initial formula θ_0 satisfies $\theta_0 \in \mathcal{B}^+(\alpha)$ (respectively, $\theta_0 \in \mathcal{B}^+(Q \setminus \alpha)$). For instance, the automaton of Figure 3 belongs to A1W$[3, \mathbf{R}]$.

6.2 Translation of LTL to A1W[2]

In the standard translation of LTL to A1W, the states of the A1W for a formula φ are subformulas of φ [35, 48]. We show that, at the price of a slightly more complicated translation, the resulting A1W for a Δ_i-formula belongs to A1W$[i]$. Thus, by the Normalization Theorem, every LTL formula can be translated into an equivalent A1W[2]. The idea of the construction is to use subformulas of the formulas as states, ensuring that

(1) transitions can only lead from a subformula to itself or to another of a smaller in the syntactic-future hierarchy (see Figure 1(b)), and
(2) accepting states are subformulas of the Π-classes of the hierarchy.

These two conditions immediately imply that the alternating automaton for a formula of Σ_i has alternation height i; indeed, every change of state involves going down in the hierarchy. There are two small technical problems. First, at the bottom of the hierarchy, we have $\Sigma_0 = \Pi_0$, and so it is not well defined whether bottom formulas are accepting states or not. Fortunately, in our automata, such states are not reachable, and so this question is irrelevant. Second, the hierarchy level of a formula is not always well defined, because some formulas do not belong to one single lowest level of the hierarchy. For example, the formula $\mathbf{X}a$ belongs to both Π_1 and Σ_1. To solve this problem, the states of the automaton will be formulas marked with one of the classes they belong to. For example, the automaton for $\mathbf{X}a$ will contain two states $(\mathbf{X}a)_{\Sigma_1}$ and $(\mathbf{X}a)_{\Pi_1}$.

Formally, we proceed as follows. A formula φ is *proper* if it is neither a Boolean constant ($\mathbf{tt}$ or $\mathbf{ff}$) nor a conjunction or a disjunction. An *atom* is a pair $\langle \varphi, \Gamma \rangle$, where φ is a proper formula, and Γ is a smallest class of the syntactic-future hierarchy such that $\varphi \in \Gamma$. A marked formula is a positive Boolean combination of atoms (including **tt** and **ff**). Given a formula φ and a class Γ such that $\varphi \in \Gamma$, we define the marked formula φ_Γ as follows: $\mathbf{tt}_\Gamma := \mathbf{tt}$, $\mathbf{ff}_\Gamma := \mathbf{ff}$, $(\varphi \vee \psi)_\Gamma := \varphi_\Gamma \vee \psi_\Gamma$, $(\varphi \wedge \psi)_\Gamma := \varphi_\Gamma \wedge \psi_\Gamma$; if φ is proper, then φ_Γ is the disjunction of all atoms $\langle \varphi, \Gamma_\varphi \rangle$ such that $\Gamma_\varphi \subseteq \Gamma$ is a smallest class containing φ. Observe, in particular, that $\varphi_\Gamma = \varphi_{\Gamma'}$ for every $\Gamma' \supseteq \Gamma$.

Example 6.5. If $\varphi = \mathbf{ff} \vee (\mathbf{X}a \wedge c)$, then $\varphi_{\Delta_1} = \mathbf{ff} \vee ((\mathbf{X}a)_{\Delta_1} \wedge c_{\Delta_1}) = (\langle \mathbf{X}a, \Sigma_1 \rangle \vee \langle (\mathbf{X}a), \Pi_1 \rangle) \wedge \langle c, \Delta_0 \rangle$.

Definition 6.6. Let $\varphi \in \Delta_i$ for some $i \geq 0$. We define the automaton $\mathcal{A}_\varphi = \langle 2^{Ap}, Q, \theta_0, \delta, \alpha \rangle$ as follows, where the symbol Γ ranges over the classes of the syntactic hierarchy:

- Q is the set of atoms $\langle \psi, \Gamma \rangle$, where ψ is a proper subformula of φ and $\Gamma \subseteq \Delta_i$.
- $\theta_0 = \varphi_{\Delta_i}$ (recall that θ_0 is a positive Boolean combination of states).
- $\alpha = \{(\psi, \Gamma) \in Q \colon \Gamma = \Pi_j \text{ for some } j \geq 0\}$.
- $\delta \colon Q \times \Sigma \rightarrow \mathcal{B}^+(Q)$ is the restriction to $Q \times \Sigma$ of a function δ that assigns to each marked formula φ_Γ (notice that, abusing language, we also call this function δ), a positive Boolean combination of states:

$$\begin{array}{rlrl}
\delta(\mathbf{tt}_\Gamma, \sigma) = & \mathbf{tt} & \delta(\mathbf{ff}_\Gamma, \sigma) = & \mathbf{ff} \\
\delta(a_\Gamma, \sigma) = & \begin{cases} \mathbf{tt} & \text{if } a \in \sigma \\ \mathbf{ff} & \text{otherwise} \end{cases} & \delta(\overline{a}_\Gamma, \sigma) = & \begin{cases} \mathbf{ff} & \text{if } a \in \sigma \\ \mathbf{tt} & \text{otherwise} \end{cases} \\
\delta((\varphi \vee \psi)_\Gamma, \sigma) = & \delta(\varphi_\Gamma, \sigma) \vee \delta(\psi_\Gamma, \sigma) & \delta((\varphi \wedge \psi)_\Gamma, \sigma) = & \delta(\varphi_\Gamma, \sigma) \wedge \delta(\psi_\Gamma, \sigma) \\
\delta((\mathbf{X}\psi)_\Gamma, \sigma) = & \psi_\Gamma & & \\
\delta((\varphi \mathbin{\mathbf{U}} \psi)_\Gamma, \sigma) = & \delta(\psi_\Gamma, \sigma) \vee \big(\delta(\varphi_\Gamma, \sigma) \wedge (\varphi \mathbin{\mathbf{U}} \psi)_\Gamma\big) & \delta((\varphi \mathbin{\mathbf{R}} \psi)_\Gamma, \sigma) = & \delta(\psi_\Gamma, \sigma) \wedge \big(\delta(\varphi_\Gamma, \sigma) \vee (\varphi \mathbin{\mathbf{R}} \psi)_\Gamma\big) \\
\delta((\varphi \mathbin{\mathbf{W}} \psi)_\Gamma, \sigma) = & \delta(\psi_\Gamma, \sigma) \vee \big(\delta(\varphi_\Gamma, \sigma) \wedge (\varphi \mathbin{\mathbf{W}} \psi)_\Gamma\big) & \delta((\varphi \mathbin{\mathbf{M}} \psi)_\Gamma, \sigma) = & \delta(\psi_\Gamma, \sigma) \wedge \big(\delta(\varphi_\Gamma, \sigma) \vee (\varphi \mathbin{\mathbf{M}} \psi)_\Gamma\big).
\end{array}$$

Example 6.7. Consider the formula $\varphi = a \mathbin{\mathbf{U}} (\mathbf{XG}(b \vee \mathbf{XF}c)) \in \Sigma_3$. The automaton $\mathcal{A}_\varphi$ is the one shown in Example 6.1, with the correspondence $q_0 = \langle \varphi, \Delta_3 \rangle$, $q_1 = \langle \mathbf{G}(b \vee \mathbf{XF}c), \Pi_2 \rangle$, and $q_2 = \langle \mathbf{F}c, \Sigma_1 \rangle$. Let us compute $\delta(q_1, \sigma)$. We get

$$\begin{aligned}
\delta(q_1, \sigma) &= \delta(\langle \mathbf{G}(b \vee \mathbf{XF}c), \Pi_2 \rangle, \sigma) = \delta((\mathbf{G}(b \vee \mathbf{XF}c))_{\Pi_2}, \sigma) \\
&= \delta((b \vee \mathbf{XF}c)_{\Pi_2}, \sigma) \wedge \langle \mathbf{G}(b \vee \mathbf{XF}c), \Pi_2 \rangle \\
&= (\delta(b_{\Pi_2}, \sigma) \vee \langle \mathbf{F}c, \Sigma_1 \rangle) \wedge \langle \mathbf{G}(b \vee \mathbf{XF}c), \Pi_2 \rangle = \begin{cases} q_1 & \text{if } b \in \sigma \\ q_1 \wedge q_2 & \text{otherwise.} \end{cases}
\end{aligned}$$

Lemma 6.8. *Let φ be a formula of Δ_i with n proper subformulas. The automaton $\mathcal{A}_\varphi$ belongs to* A1W$[i]$, *has $2n$ states, and recognizes $\mathcal{L}(\varphi)$.*

Proof. Let us first show that $\mathcal{A}_\varphi$ belongs to A1W$[i]$. It follows immediately from the definition of $\mathcal{A}_\varphi$ that for every two states $\langle \psi, \Gamma \rangle, \langle \psi', \Gamma' \rangle$ of $\mathcal{A}_\varphi$, if $\langle \psi, \Gamma \rangle \longrightarrow \langle \psi', \Gamma' \rangle$, then $\Gamma' \subseteq \Gamma$. So in every path, there are at most $(i-1)$ alternations between Σ and Π classes. Since the states of α are those annotated with Π classes, there are also at most $(i-1)$ alternations between α and non-α states in a path.

To show that $\mathcal{A}_\varphi$ has at most $2n$ states, observe that for every formula ψ there are at most two smallest classes of the syntactic-future hierarchy containing ψ. So $\mathcal{A}_\varphi$ has at most two states for each proper subformula of φ.

To prove that $\mathcal{A}_\varphi$ recognizes $\mathcal{L}(\varphi)$ one shows by induction on ψ that $\mathcal{A}_\psi$ recognizes $\mathcal{L}(\psi)$ from every marked formula ψ_Γ. The proof is completely analogous to the one given in the work of Vardi [48]. □

6.3 Determinization of AWW[2]

We present a determinization procedure for AWW[2, **R**] and AWW[2, **A**] inspired by the breakpoint construction from Miyano and Hayashi [34]. More precisely, given an AWW[2, **R**], we construct an equivalent deterministic co-Büchi automaton, and given an AWW[2, **A**], we construct an equivalent deterministic Büchi automaton. We only describe the construction for AWW[2, **R**], as the one for AWW[2, **A**] is dual.

The section is structured as follows. First, we introduce the notion of the level sequence of a run. Second, we present the fundamental property of AWW[2, **R**] that the procedure will exploit. Third, we describe the procedure itself. Finally, we combine the procedures for AWW[2, **R**] and AWW[2, **A**] into a procedure that, given an arbitrary AWW[2], constructs an equivalent deterministic Rabin automaton.

6.3.1 Level Sequence of a Run. Let $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$ be an alternating Büchi automaton. A set $U \subseteq Q$ of states is called a *level*. If $U \subseteq \alpha$, then U is an *α-level*. Given two levels U, U' and a letter $a \in \Sigma$, we say that U' is a *successor of U w.r.t. $a \in \Sigma$*, also called an *a-successor* of U, if for every $q \in U$ there is a minimal model S_q of $\delta(q, a)$ such that $U' = \bigcup_{q \in U} S_q$. We make two observations:

- The empty set of states is a level, and moreover an α-level for any $\alpha \subseteq Q$. The empty level has exactly one a-successor for every $a \in \Sigma$, namely the empty level itself.
- A level U has no a-successors iff it contains a state q such that $\delta(q, a) = \mathbf{ff}$. Indeed, if $\delta(q, a) = \mathbf{ff}$, then there is no minimal model of $\delta(q, a)$. Conversely, if $\delta(q, a) \neq \mathbf{ff}$ for every $q \in U$, then $\delta(q, a)$ has at least one minimal model S_q for every state $q \in U$, and the set $U' := \bigcup_{q \in U} S_q$ is a (possibly empty) a-successor of U.

Recall that, given a run $G = (V, E)$ on a word w, we define for every $l \geq 0$ the set $Q_l := \{q \colon (q, l) \in V\}$. By the definitions of run and level, for every $l \geq 0$, either $\delta(q, a) = \mathbf{ff}$ for some $q \in Q_l$ or Q_{l+1} is a $w[l]$-successor of Q_l. We define the *level sequence* of G as a certain prefix of the infinite sequence $Q_0\, Q_1\, Q_2 \ldots$:

- If there exists a smallest $l \geq 0$ such that Q_l has no $w[l]$-successor, then the level sequence is the finite sequence $Q_0\, Q_1 \cdots Q_l$.
- Otherwise, the level sequence of G is the infinite sequence $Q_0\, Q_1\, Q_2 \cdots$ itself.

Example 6.9. Let us compute the level sequences of the four runs of Figure 2:

- *First run*: $\{q_0\}\, \{q_1\}\, \{q_1, q_2\}^\omega$. Since $\alpha = \{q_1\}$, the only α-level of the level sequence is $\{q_1\}$.
- *Second run*: $\{q_0\}\, \{q_0\}\, \{q_0\}$. The level sequence ends at the second level because $\delta(q_0, \{c\}) = \mathbf{ff}$.
- *Third run*: $\{q_0\}\, \{q_1\}\, \{q_1, q_2\}^\omega$. Observe that although the first and third runs have the same sequence of levels, the first run is accepting but the second one is not.
- *Fourth run*: $\{q_0\}\, \{q_0\}\, \emptyset^\omega$. Indeed, $\emptyset$ is a c-successor of $\{q_0\}$ because $\delta(q_0, c) = \mathbf{tt}$, and so $\emptyset$ is a minimal model of $\delta(q_0, c)$. While the run itself is finite, its level sequence is infinite.

6.3.2 Fundamental Property of AWW[2, **R**]. The first and third runs of Example 6.9 have the same level sequence, but one is accepting and the other is not. Therefore, the level sequence of a run does not determine in general whether the run is accepting. However, the runs of Example 6.9 are runs of an automaton in AWW[3, **R**]. The following lemma shows that for AWW[2, **R**], the level sequence does determine acceptance; in other words, to decide whether a run is accepting or not, it suffices to examine its level sequence.

Lemma 6.10. *Let $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$ be an* AWW[2, **R**]. *A run $G = (V, E)$ of $\mathcal{A}$ on a word w is accepting iff*

(a) the level sequence of G is the infinite sequence $Q_0\, Q_1\, Q_2 \cdots$, and
(b) there is a threshold $k \geq 0$ such that Q_l is an α-level for every $l \geq k$.

Proof. We first observe that the level sequence of G satisfies the following property for every $l \geq 0$: if the sets Q_l and Q_{l+1} belong to the sequence, then Q_{l+1} is a $w[l]$-successor of Q_l. Indeed, if Q_{l+1} belongs to the level sequence, then $\delta(q, a) \neq \mathbf{ff}$ for every $q \in Q_l$ and so, by the definition of a run, the set Q_{l+1} is a $w[l]$-successor of Q_l. Now we prove the two directions of the lemma.

($\Rightarrow$): Let $G = (V, E)$ be an accepting run of $\mathcal{A}$ on w. By the definition of acceptance, we have $\delta(q, w[l]) \neq \mathbf{ff}$ for every $l \geq 0$ and $q \in Q_l$, and so, by the preceding observation, Q_{l+1} is a $w[l]$-successor of Q_l. By the definition of the level sequence of a run, the level sequence of G is infinite. This proves (a).

Let us now prove (b). Say that a node $(q, l) \in V$ is accepting if $q \in \alpha$, and rejecting otherwise. We prove two claims.

Claim 1. *All descendants of an accepting node of G are also accepting.*

Proof. Since $\mathcal{A}$ is an AWW$[2, \mathbf{R}]$, the initial formula θ_0 of $\mathcal{A}$ satisfies $\theta_0 \in \mathcal{B}^+(Q \setminus \alpha)$ by Definition 6.4, and so every node $(q, 0)$ of G is rejecting. So every node of G is a descendant of a rejecting node. Since $\mathcal{A}$ has alternation height 2, every path of G has at most *one* alternation of accepting and rejecting nodes. Therefore, no descendant of an accepting node is rejecting. □

Claim 2. *The set $R \subseteq V$ of rejecting nodes of V is finite.*

Proof. Assume that R is infinite. Let $G_R = (R, E \cap (R \times R))$, and let $(q, l) \in R$. If $l \geq 1$, then by the definition of a run, (q, l) has at least one predecessor in G. By Claim 1, all predecessors of a rejecting node are rejecting, and so (q, l) also has at least one predecessor in G_R. It follows that the only nodes of G_R without predecessors are those of the form $(q, 0)$, and so finitely many. Since R is infinite and G_R is finitely branching, G_R contains an infinite path by König's lemma. By the definition of G_R, this path only contains non-accepting nodes, contradicting that G is an accepting run. This proves the claim. □

By Claim 2, there exists a threshold k such that $k \geq m$ for every $(q, m) \in R$. It follows that $(q, l) \in V$ is an accepting node for every $l \geq k$, and so that Q_l is an α-level for every $l \geq k$.

($\Leftarrow$): We prove this direction for arbitrary alternating automata. Let $G = (V, E)$ be a run satisfying (a) and (b). We show that it also satisfies the two properties of an accepting run:

- $\delta(q, w[l]) \not\equiv \mathbf{ff}$ for every $l \geq 0$ and every $q \in Q_l$.
  By (a), the level sequence of G is infinite, and so, by the definition of the level sequence of a run, we are done.
- Every infinite path of G contains infinitely many accepting nodes.
  Every infinite path of G contains exactly one node of the form (q, l) for every $l \geq 0$. By (b), the node (q, l) of the path is accepting for every $l \geq k$. □

6.3.3 A Determinization Procedure for AWW$[2, \mathbf{R}]$. Let $\mathcal{A}$ be an AWW$[2, \mathbf{R}]$ with set of states Q and $|Q| = n$. We construct an equivalent deterministic co-Büchi automaton $\mathcal{D}$ whose states are pairs (*Levels*, *Promising*), where *Levels* $\subseteq 2^Q$ (recall that a level of an AWW is a set of states) and *Promising* $\subseteq 2^\alpha \cap$ *Levels*. Since there exist 3^{2^n} pairs satisfying these conditions, $\mathcal{D}$ has at most 3^{2^n} states.

Let us first give an informal but hopefully intuitive description of the transitions and acceptance condition of the deterministic co-Büchi automaton. The transitions of $\mathcal{D}$ are chosen to ensure that, after reading a finite word $w_{0i} = a_0 \dots a_i$, the automaton is in the state $(\mathit{Levels}_i, \mathit{Promising}_i)$, where

- $Levels_i$ contains the i-th levels of all runs of $\mathcal{A}$ on all words having w_{0i} as prefix (when they exist), and
- $Promising_i \subseteq Levels_i$ contains some α-levels of $Levels_i$. These levels are "promising," intuitively meaning that they could belong to the infinite tail of α-levels of the level sequence of an accepting run.

For this, when $\mathcal{D}$ reads a_{i+1}, it moves from $(Levels_i, Promising_i)$ to $(Levels_{i+1}, Promising_{i+1})$, where $Levels_{i+1}$ contains all the a_{i+1}-successors of the levels of $Levels_i$, and $Promising_{i+1}$ is defined as follows:

- If $Promising_i \neq \emptyset$, then $Promising_{i+1}$ contains the a_{i+1}-successors of $Promising_i$. (Recall that $\mathcal{A}$ is an AWW$[2, \mathbf{R}]$, and so if $Promising_i$ contains α-levels, then so does $Promising_{i+1}$.)
- If $Promising_i = \emptyset$, then $Promising_{i+1}$ contains all α-levels of $Levels_{i+1}$.

Finally, the co-Büchi condition contains the states $(Levels, Promising)$ such that $Promising = \emptyset$.

Intuitively, during its run on a word w, the automaton $\mathcal{D}$ tracks the promising levels, removing those without successors, because they cannot belong to an accepting run. If some run G of $\mathcal{A}$ accepts w, then by Lemma 6.10 the level sequence $Q_0\, Q_1\, Q_2 \cdots$ of G is infinite, and there is $k \geq 0$ such that the levels $Q_k, Q_{k+1}, Q_{k+2} \cdots$ are all promising. So the sets $Promising_k, Promising_{k+1}, Promising_{k+2} \cdots$ are all non-empty, and $\mathcal{D}$ accepts. Conversely, assume there is k such that $Promising_k, Promising_{k+1}, Promising_{k+2} \cdots$ are all non-empty. Then we can construct a run of $\mathcal{A}$ on w by picking a sequence of levels $U_k\, U_{k+1}\, U_{k+2} \cdots$ such that $U_k \in Promising_k$ and U_l is a $w[l]$-successor of U_{l-1} for every $l > k$, which belong by definition to $Promising_l$, and then picking $U_0\, U_1 \cdots U_{k-1}$ such that $U_l \in Levels_l$ for every $0 \leq l \leq k-1$. The level sequence of this run is infinite, and from U_k onward it only contains α-levels, which implies that the run is accepting.

Lemma 6.11. *For every $\mathcal{A} \in$ AWW$[2, \mathbf{R}]$ and infinite word w on Σ, if there is $k \geq 0$ such that $Promising_i \neq \emptyset$ for all $i \geq k$, there is an infinite sequence $U_k\, U_{k+1} \cdots U_i\, U_{i+1} \cdots$ such that $U_i \in Promising_i$ and U_{i+1} is a $w[i+1]$-successor of U_i for all $i \geq k$.*

Proof. Let $F = (V, E)$ be the graph with vertices $V = \bigcup_{i \geq k}\{(U_i, i) \mid U_i \in Promising_i\}$ and edges $E = \bigcup_{i \geq k}\{((U, i), (U', i+1)) \mid U \in Promising_i$ and U' is a $w[i]$-successor of $U\}$. This is well defined since such a U' belongs to $Promising_{i+1}$ by definition. V is infinite because $Promising_i \neq \emptyset$ for all $i \geq k$, and the degree of each vertex is finite since $|Promising_i| \leq 2^{|Q|}$. Moreover, F is acyclic, because the second entry of the pair is monotonically increasing and consists of as many connected components as $U_k \in Promising_k$. Indeed, $U_{i+1} \in Promising_{i+1}$ is a $w[i+1]$-successor of some $U_i \in Promising_i$ by definition of the $Promising$ sets, so inductively, we can go back to a set $U_k \in Promising_k$. Hence, F is the union of $|Promising_k| \leq 2^{|Q|}$ trees, and at least one of them must be infinite because V is.

Let T be an infinite tree of F. Since it is both finite and finitary, by the König lemma, there is an infinite branch $(U_k, k), (U_{k+1}, k+1), \ldots$. We can then extract the infinite sequence $U_k\, U_{k+1} \cdots$ satisfying the conditions in the statement of the lemma. □

For the formal definition of $\mathcal{D}$, it is convenient to identify subsets of 2^Q and 2^α with formulas of $\mathcal{B}^+(Q)$, $\mathcal{B}^+(\alpha)$ (i.e., we identify a formula and its set of models). Further, we lift $\delta\colon Q \times \Sigma \mapsto \mathcal{B}^+(Q)$ to $\delta\colon \mathcal{B}^+(Q) \times \Sigma \mapsto \mathcal{B}^+(Q)$ in the canonical way. Finally, given $\varphi \in \mathcal{B}^+(Q)$ and $S \subseteq Q$, we let $\varphi[\mathbf{ff}/S]$ denote the result of substituting $\mathbf{ff}$ for every state of S in φ. With these notations, the deterministic Büchi automaton $\mathcal{D}$ equivalent to $\mathcal{A}$ can be described in four lines: $\mathcal{D} = \langle \Sigma, Q', q'_0, \delta', \alpha' \rangle$, where

$Q' = \mathcal{B}^+(Q) \times \mathcal{B}^+(\alpha)$, $q_0' = (\theta_0, \mathbf{ff})$, $\alpha' = \{(\theta, \mathbf{ff})\colon \theta \in \mathcal{B}^+(Q)\}$, and

$$\delta'((q,p),a) = \begin{cases} (\delta(q,a), \delta(p,a)) & \text{if } p \not\equiv \mathbf{ff} \\ (\delta(q,a), \delta(q,a)[\mathbf{ff}/Q \setminus \alpha]) & \text{otherwise.} \end{cases}$$

Lemma 6.12. *For every $\mathcal{A} \in$ AWW$[2, \mathbf{R}]$ with n states, the deterministic co-Büchi automaton $\mathcal{D}$ defined previously satisfies $L(\mathcal{A}) = L(\mathcal{D})$ and has 3^{2^n} states. Dually, for every $\mathcal{A}' \in$ AWW$[2, \mathbf{A}]$ with n' states, there exists a deterministic Büchi automaton $\mathcal{D}'$ that has $3^{2^{n'}}$ states and that satisfies $L(\mathcal{A}') = L(\mathcal{D}')$.*

Proof. Assume $\mathcal{A}$ accepts a word w. Let $G = (V, E)$ be an accepting run of $\mathcal{A}$ on w. Since $\mathcal{A} \in$ AWW$[2, \mathbf{R}]$, we can apply Lemma 6.10, and so there exists an index k such that all levels of G after the k-th one are α-levels and have at least one successor. It follows that the run $(\mathit{Levels}_0, \mathit{Promising}_0), (\mathit{Levels}_1, \mathit{Promising}_1) \ldots$ of $\mathcal{D}$ on w satisfies $\mathit{Promising}_l \neq \emptyset$ for every $l \geq k$, and so $\mathcal{D}$ accepts.

Assume $\mathcal{D}$ accepts a word w. Let $(\mathit{Levels}_0, \mathit{Promising}_0), (\mathit{Levels}_1, \mathit{Promising}_1) \ldots$ be the run of $\mathcal{D}$ on w. By the definition of the acceptance condition of $\mathcal{D}$, there is $k \geq 0$ such that $\mathit{Promising}_i \neq \emptyset$ for every $i \geq k$. By Lemma 6.11, there is an infinite sequence $U_k\, U_{k+1} \cdots U_i \cdots$ such that $U_i \in \mathit{Promising}_i$ and U_{i+1} is a $w[i+1]$-successor of U_i for every $i \geq k$. Then, choose levels $U_0\, U_1 \cdots U_{k-1}$ such that for every $1 \leq l \leq k$, the level U_{l-1} is a predecessor of U_l that belongs to Levels_{l-1} (i.e., U_l is a $w[l]$-successor of U_{l-1}; this is always possible by the definition of δ'). Let $G = (V, E)$ be the graph given by

- for every $l \geq 0$, $(q, l) \in V$ iff $q \in U_l$, and
- $((q, l), (q', l+1)) \in E$ iff $q \in U_l$ and $q' \in S_q$, where S_q is a minimal model of $\delta(q, w[l])$ satisfying $U_{l+1} = \bigcup_{q \in U_l} S_q$, which exists by definition.

Since $U_k \in \mathit{Promising}_k$, and promising sets only contain α-levels, U_k is an α-level. Since $\mathcal{A} \in$ AWW$[2, \mathbf{R}]$, successors of α-levels are also α-levels, and so all of $U_k, U_{k+1}, U_{k+2}, \ldots$ are α-levels. By Lemma 6.10, G is an accepting run of $\mathcal{A}$.

The dual part of the lemma is proven by complementing $\mathcal{A}'$, applying the construction we have just described, and replacing the co-Büchi condition by a Büchi condition. □

6.3.4 *A Determinization Procedure for* AWW[2].

Lemma 6.12 leads to a determinization procedure for AWW[2]. Combining the procedures for AWW$[2, \mathbf{R}]$ and AWW$[2, \mathbf{A}]$, we show how to construct for every automaton in AWW[2] an equivalent deterministic Rabin automata.

Rabin automata generalize both Büchi and co-Büchi automata. A Rabin automaton over an alphabet Σ is a tuple $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$, where Σ, Q, q_0, δ are defined as for Büchi or co-Büchi automata, and α is a set of *Rabin pairs* $(F, I) \subseteq Q \times Q$. Deterministic Rabin automata are defined as for Büchi or co-Büchi automata (the definitions do not involve the accepting condition α). The same applies to the runs of a Rabin automaton. The only new definition is that of an accepting run. A run $G = (V, E)$ of a Rabin automaton is *accepting* if

(a) $\delta(q, w[l]) \not\equiv \mathbf{ff}$ for every $q \in Q_l$, and
(b) for every infinite path of G there exists a Rabin pair $(F, I) \in \alpha$ such that the path visits infinitely often nodes $(q, l) \in V$ such that $q \in I$, and only finitely often nodes $(q, l) \in V$ such that $q \in F$.

We abbreviate deterministic Rabin automaton to DRW.

Lemma 6.13. *For every $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle \in$ AWW[2] with $n = |Q|$ states and $m = |\mathcal{M}_{\theta_0}|$ minimal models of θ_0 there exists an equivalent DRW $\mathcal{D}$ with $2^{2^{n + \log_2 m + 2}}$ states and with m Rabin pairs.*

Proof. Let $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$. Given $Q' \subseteq Q$, let $\mathcal{A}_{Q'}$ be the AWW[2] obtaining from $\mathcal{A}$ by substituting $\bigwedge_{q \in Q'} q$ for the initial formula θ_0. We claim that for each minimal model $S \in \mathcal{M}_{\theta_0}$, we can construct a deterministic Rabin automaton (DRW) $\mathcal{D}_S$ with at most $2^{2^{n+2}}$ states and a single Rabin pair, recognizing the same language as $\mathcal{A}_S$. Let us first see how to construct $\mathcal{D}$, assuming the claim holds. By the claim, we have $\mathcal{L}(\mathcal{A}) = \bigcup_{S \in \mathcal{M}_{\theta_0}} \mathcal{L}(\mathcal{A}_S)$. So we define $\mathcal{D}$ as the union of all of the automata $\mathcal{D}_S$. Recall that given two DRWs with n_1, n_2 states and p_1, p_2 Rabin pairs, we can construct a DRW for the union of their languages with $n_1 \times n_2$ states and $p_1 + p_2$ pairs. Since θ_0 has m models, $\mathcal{D}$ has at most m Rabin pairs and $\left(2^{2^{n+2}}\right)^m = 2^{2^{n+\log_2 m+2}}$ states.

It remains to prove the claim. Partition S into $S \cap \alpha$ and $S \setminus \alpha$. We have $\mathcal{A}_{S\cap\alpha} \in$ AWW[2, **A**] and $\mathcal{A}_{S\setminus\alpha} \in$ AWW[2, **R**]. By Lemma 6.12, there exists a deterministic Büchi automaton $\mathcal{D}_{S\cap\alpha}$ and a deterministic co-Büchi automaton $\mathcal{D}_{S\setminus\alpha}$ equivalent to $\mathcal{A}_{S\cap\alpha}$ and $\mathcal{A}_{S\setminus\alpha}$, respectively, both with at most 3^{2^n} states. Intersecting these two automata yields a deterministic Rabin automaton with at most $3^{2^{n+1}} \leq 2^{2^{n+2}}$ states and a single Rabin pair, and we are done. □

Remark 6.14. The construction of Lemma 6.12 is close to the translation by Miyano and Hayashi [35] of alternating automata to non-deterministic automata, and to the translation by Schneider [45, p. 219] of Σ_2 formulas to deterministic co-Büchi automata, all based on the break-point idea.

6.4 Translation of LTL to DRW

Combining a normalization procedure LTL→ Δ_2, the procedure Δ_2 → A1W[2] of Section 6.2 and the determinization procedure of Section 6.3, we obtain a translation LTL→DRW. Since the procedures involve a single-exponential, a linear, and a double-exponential blow-up, respectively, a straightforward composition only yields a triple-exponential bound. However, a closer examination of the closed-form Δ_2-formula provided by Theorem 4.22 allows us to reduce the bound to double exponential.

Given a formula φ, by Theorem 4.22, we have

$$\varphi \equiv \bigvee_{\substack{M \subseteq B_{\mathrm{GF}} \\ N \subseteq B_{\mathrm{FG}}}} \varphi_{M,N} \quad \text{where} \quad \varphi_{M,N} := \left(\varphi\langle M \rangle \wedge \bigwedge_{\mathbf{FG}\psi \in N} \mathbf{FG}(\psi\langle\langle M \rangle\rangle) \wedge \bigwedge_{\mathbf{GF}\psi \in M} \mathbf{GF}(\psi[\![N]\!]) \right). \tag{10}$$

The key result is that each $\varphi_{M,N}$ has a small number of different proper subformulas. (In fact, the number is even linear in the number of subformulas of φ.) Invoking Lemma 6.8, we obtain an A1W[2] with $O(|sf(\varphi)|)$ states that recognizes $\mathcal{L}(\varphi_{M,N})$.

Lemma 6.15. *Let φ be a formula. For every $M \subseteq B_{\mathbf{GF}}$ and $N \subseteq B_{\mathbf{FG}}$, there exists an* A1W[2] *with $O(|sf(\varphi)|)$ states that recognizes $\mathcal{L}(\varphi_{M,N})$.*

Proof. By Lemma 6.8, some A1W[2] with $O(|sf(\varphi_{M,N})|)$ states recognizes $\mathcal{L}(\varphi_{M,N})$. So it suffices to show that $|sf(\varphi_{M,N})| \in O(|sf(\varphi)|)$, but this is Lemma 4.24. □

Theorem 6.16. *Let φ be a formula with n proper subformulas. Let $\mathcal{A}_\varphi$ be the DRW obtained by*

- *normalizing φ into $\bigvee_{M \subseteq B_{\mathrm{GF}}, N \subseteq B_{\mathrm{FG}}} \varphi_{M,N}$ as in Equation (10),*
- *constructing a* A1W[2] *$\mathcal{A}_{M,N}$ for each $\varphi_{M,N}$ as in Definition 6.6,*
- *transforming each $\mathcal{A}_{M,N}$ into an equivalent DRW $\mathcal{A}'_{M,N}$ as in Lemma 6.13, and*
- *constructing a DRW recognizing the union of the languages of all $\mathcal{A}'_{M,N}$.*

The DRW $\mathcal{A}_\varphi$ recognizes $\mathcal{L}(\varphi)$, and has $2^{2^{O(n^2)}}$ states and 2^n Rabin pairs.

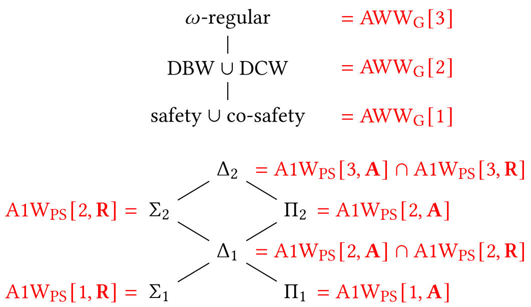

ω-regular
$= \mathrm{AWW_G}[3]$
DBW ∪ DCW
$= \mathrm{AWW_G}[2]$
safety ∪ co-safety
$= \mathrm{AWW_G}[1]$
Δ_2
$= \mathrm{A1W_{PS}}[3, \mathbf{A}] \cap \mathrm{A1W_{PS}}[3, \mathbf{R}]$
$\mathrm{A1W_{PS}}[2, \mathbf{R}] = \Sigma_2$
$\Pi_2 = \mathrm{A1W_{PS}}[2, \mathbf{A}]$
Δ_1
$= \mathrm{A1W_{PS}}[2, \mathbf{A}] \cap \mathrm{A1W_{PS}}[2, \mathbf{R}]$
$\mathrm{A1W_{PS}}[1, \mathbf{R}] = \Sigma_1$
$\Pi_1 = \mathrm{A1W_{PS}}[1, \mathbf{A}]$

Fig. 4. Expressive power of AWWs after Gurumurthy et al. [19], and of A1Ws after Pelánek and Strejcek [38].

Proof. Due to Lemma 6.15, the alternating automaton $\mathcal{A}_{M,N}$ that recognizes $\mathcal{L}(\varphi_{M,N})$ belongs to A1W[2] and has $O(n)$ states. Applying the construction of Lemma 6.13, we obtain a DRW with $2^{2^{O(n)}}$ states and a single Rabin pair. Using the union operation for DRWs, we obtain a DRW for φ with $(2^{2^{O(n)}})^{2^{O(n)}} = 2^{2^{O(n)}}$ states and 2^n Rabin pairs. □

6.5 Determinization of Lower Classes

We now determinize AWW[1]. A deterministic automaton is *terminal-accepting* if all states are rejecting except a single accepting sink with a self-loop, and *terminal-rejecting* if all states are accepting except a single rejecting sink with a self-loop. It is easy to see that terminal-accepting and terminal-rejecting deterministic automata are closed under union and intersection. When applied to AWW$[1, \mathbf{A}]$, the construction of Lemma 6.12, yields automata whose states have a trivial *Promising* set (either the empty set or the complete level). Further, the successor of an α-level is also an α-level. From these observations we easily get the following corollary.

Corollary 6.17. *Let $\mathcal{A}$ be an automaton with n states:*

- *If $\mathcal{A} \in$ AWW$[1, \mathbf{R}]$ (respectively, $\mathcal{A} \in$ AWW$[1, \mathbf{A}]$), then there exists a deterministic terminal-accepting (respectively, terminal-rejecting) automaton recognizing $\mathcal{L}(\mathcal{A})$ with 2^{2^n} states.*
- *If $\mathcal{A} \in$ AWW[1], then there exists deterministic weak automaton recognizing $\mathcal{L}(\mathcal{A})$ with $2^{2^{n+\log_2 |M_{\theta_0}|+1}}$ states.*

7 A HIERARCHY OF ALTERNATING WEAK AND VERY WEAK AUTOMATA

The expressive power of weak and very weak alternating automata has been studied by Gurumurthy et al. [19] and by Pelánek and Strejcek [38], respectively. Both of these works identify the number of alternations between accepting and non-accepting states as an important parameter, and define a hierarchy of automata classes based on it. Let $\mathrm{AWW_G}[k]$ denote the class of AWW with at most $(k-1)$ alternations defined by Gurumurthy et al. [19]. Similarly, let $\mathrm{A1W_{PS}}[k, \mathbf{A}]$ and $\mathrm{A1W_{PS}}[k, \mathbf{R}]$ denote the classes of A1W with at most $(k-1)$ alternations and accepting or non-accepting initial state, respectively, defined by Pelánek and Strejcek [38]. Finally, define $\mathrm{A1W_{PS}}[k] = \mathrm{A1W_{PS}}[k, \mathbf{A}] \cup \mathrm{A1W_{PS}}[k, \mathbf{R}]$.[4] Figure 4 shows the results of Gurumurthy et al. [19] and Pelánek and Strejcek [38]. We abuse language, and, for example, write $\Pi_2 = \mathrm{A1W_{PS}}[2, \mathbf{A}]$ to denote that the class of languages satisfying formulas in Π_2 and the class of languages recognized by automata in $\mathrm{A1W_{PS}}[2, \mathbf{A}]$ coincide.

Unfortunately, the results of Gurumurthy et al. [19] and Pelánek and Strejcek [38] do not "match." Due to slight differences in the definitions of height, such as the treatment of $\delta(\cdot) = \mathbf{ff}$ and $\delta(\cdot) = \mathbf{tt}$,

[4] In the work of Pelánek and Strejcek [38], the classes have different names.

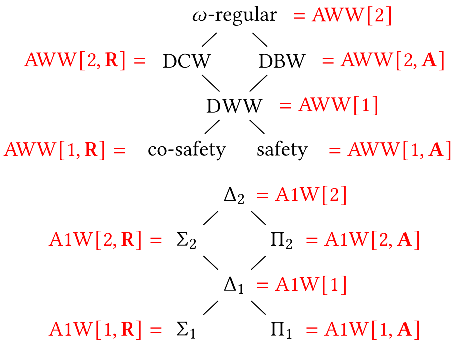


Fig. 5. Expressive power of AWWs and A1Ws.

the restriction to very weak automata of $\mathrm{AWW}_{\mathrm{G}}[k]$ does not match any class $\mathrm{A1W}_{\mathrm{PS}}[k']$ (i.e., $\mathrm{AWW}_{\mathrm{G}}[k] \cap \mathrm{A1W} \neq \mathrm{A1W}_{\mathrm{PS}}[k']$) and, vice versa, extending $\mathrm{A1W}_{\mathrm{PS}}[k]$ does not yield any $\mathrm{AWW}_{\mathrm{G}}$ $[k']$. We show that our new definition of height unifies the two hierarchies, yielding the pleasant result shown in Figure 5.

Proposition 7.1. *We have the following:*

(1) $\mathrm{AWW}[2] = \omega$*-regular,* $\mathrm{AWW}[2, \mathbf{A}] = \mathrm{DBW}$*, and* $\mathrm{AWW}[2, \mathbf{R}] = \mathrm{DCW}$.

(2) $\mathrm{AWW}[1] = \mathrm{DWW}$, $\mathrm{AWW}[1, \mathbf{A}] =$ *safety, and* $\mathrm{AWW}[1, \mathbf{R}] =$ *co-safety.*

(3) For $i \in \{1, 2\}$, $\mathrm{A1W}[i] = \Delta_i$, $\mathrm{A1W}[i, \mathbf{R}] = \Sigma_i$, *and* $\mathrm{A1W}[i, \mathbf{A}] = \Pi_i$.

Proof. We sketch the proof of (1) and (3). The proof of (2) is analogous to that of (1).

(1) The $\subseteq$-inclusion follows immediately from Lemmas 6.12 and 6.13 and Corollary 6.17. The $\supseteq$-inclusion is a slight adaptation of similar proofs in the work of Gurumurthy et al. [19]. To translate a DCW into a $\mathrm{AWW}[2, \mathbf{R}]$, we duplicate the set of states into two sets of marked and unmarked states. We remove from the marked states all rejecting states, and add transitions that allow unmarked states to nondeterministically choose to move to another unmarked state, or to its marked copy. Finally, we define all unmarked states to be rejecting and all marked states to be accepting. The proof of $\mathrm{AWW}[2, \mathbf{A}] \supseteq \mathrm{DBW}$ is dual. Finally, the inclusion $\mathrm{AWW}[2] \supseteq \omega$-regular follows from the previous two results; indeed, every ω-regular language is recognized by a DRW [43], and every DRW is equivalent to a Boolean combination of DBWs and DCWs, which we can express in the initial formula θ_0 of the $\mathrm{AWW}[2]$.

(3) The $\supseteq$-inclusion for Δ_i is proven in Lemma 6.8. For a formula φ that belongs to Σ_i (Π_i), we also rely on Lemma 6.8, but add a new initial state, φ_{Σ_i} (φ_{Π_i}), that is marked as rejecting (accepting) such that the automaton belongs to $\mathrm{A1W}[i, \mathbf{R}]$ ($\mathrm{A1W}[i, \mathbf{A}]$).

For the $\subseteq$-inclusion, let $\mathcal{A} = \langle \Sigma, Q, \theta_0, \delta, \alpha \rangle$ be a very weak alternating automaton with $\Sigma = 2^{Ap}$. We use the translation from A1W to LTL presented in the work of Löding and Thomas [25, Theorem 6], with minimal modifications, to define a formula $\psi_{\mathcal{A}}$ such that $\mathcal{L}(\psi_{\mathcal{A}}) = \mathcal{L}(\mathcal{A})$. Then, we show that when $\mathcal{A}$ belongs to one of the classes in the hierarchy, $\psi_{\mathcal{A}}$ belongs to the corresponding class of formulas. For the proof of correctness of the translation, we refer the reader to the work of Löding and Thomas [25].

For the definition of $\psi_{\mathcal{A}}$, we assign to every $\theta \in \mathcal{B}^+(Q)$ an LTL formula $\psi(\theta)$ such that $\mathcal{L}(\psi(\theta)) = \mathcal{L}(\mathcal{A}_\theta)$, where $\mathcal{A}_\theta$ denotes $\mathcal{A}$ with θ as initial formula, and set $\psi_{\mathcal{A}} := \psi(\theta_0)$. Similarly, for the definition of $\psi(\theta)$, we first assign a formula $\psi(q)$ to every state q, and then define $\psi(\theta)$ as the result of substituting $\psi(q)$ for q in θ, for every state q. It remains to define $\psi(q)$. Using that $\mathcal{A}$ is very weak, we proceed inductively—that is, we assume that $\psi(q')$ has already been defined for all q'

such that $q \longrightarrow q'$ and $q \neq q'$. For every $q \in Q$ and $\sigma \in 2^{Ap}$, let $\theta_{q,\sigma}$ and $\theta'_{q,\sigma}$ be formulas such that $\delta(q,\sigma) \equiv (q \wedge \theta_{q,\sigma}) \vee \theta'_{q,\sigma}$ (it is easy to see that they exist). Define

$$\psi(q) := \begin{cases} \varphi_q \mathbf{U} \varphi'_q & \text{if } q \notin \alpha \\ \varphi_q \mathbf{W} \varphi'_q & \text{if } q \in \alpha, \end{cases}$$

where

$$\varphi_q := \bigvee_{\sigma \subseteq \Sigma} \left(\bigwedge_{a \in \sigma} a \wedge \bigwedge_{a \notin \sigma} \neg a \wedge \mathbf{X} \psi(\theta_{q,\sigma}) \right) \qquad \varphi'_q := \bigvee_{\sigma \subseteq \Sigma} \left(\bigwedge_{a \in \sigma} a \wedge \bigwedge_{a \notin \sigma} \neg a \wedge \mathbf{X} \psi(\theta'_{q,\sigma}) \right) .$$

Since this translation assigns to each **U**-formula a rejecting state and to each **W**-formula an accepting state, the syntax tree of $\psi_{\mathcal{A}}$ has an alternation between **U** and **W** exactly when there is an alternation between accepting and non-accepting states. This yields all of the desired inclusions in $\Sigma_1, \Pi_1, \ldots, \Delta_2$. □

Moreover, our single exponential normalization procedure for LTL transfers to a single exponential normalization procedure for A1W.

Lemma 7.2. *Let $\mathcal{A}$ be an* A1W *with n states over an alphabet with m letters. There exists $\mathcal{A}' \in$* A1W[2] *with $2^{O(nm)}$ states such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Proof. The translation from A1W to LTL used in Proposition 7.1 (an adaption of Löding and Thomas [25]) yields a formula $\psi_{\mathcal{A}}$ with at most $O(mn)$ proper subformulas. Applying our normalization procedure to $\psi_{\mathcal{A}}$ yields an equivalent formula in Δ_2 with at most $2^{O(mn)}$ proper subformulas (Lemma 6.15). Applying Lemma 6.8, we obtain the postulated automaton $\mathcal{A}'$. □

8 CONCLUSION

We have presented two purely syntactic normalization procedures for LTL that transform a given formula into an equivalent formula in Δ_2—that is, a formula with at most one alternation between least- and greatest-fixpoint operators. The procedure has single exponential blow-up, improving on the prohibitive non-elementary cost of previous constructions. The much better complexity of the new procedure (recall that normalization procedures for CNF and DNF are also exponential) makes it attractive for its implementation and use in tools. We have presented a first promising application, namely a novel translation from LTL to DRW with double-exponential blow-up. Finally, we have shown that the normalization procedure for LTL can be transferred to a normalization procedure for very weak alternating automata.

We believe that these results demystify the Normalization Theorem of Chang, Manna, and Pnueli, which heavily relied on automata-theoretic results, and involved a non-elementary blowup. Indeed, the only conceptual difference between our rewrite system and the one for bringing Boolean formulas in CNF is the use of rewrite rules with contexts.

Our normalization procedure has already found applications to the translation of LTL formulas into deterministic or limit-deterministic ω-automata [22, 33]. Until now, normalization had not been considered, because of the non-elementary blow-up, much higher than the double-exponential blow-up of existing constructions. With our new procedure, translations that first normalize the formula, and then apply efficient formula-to-automaton procedures specifically designed for formulas in normal form, have become competitive. Our system of rewriting rules makes this even more attractive. More generally, we believe that the design of analysis procedures for formulas in normal form (to check satisfiability, equivalence, or other properties) should be further studied in the coming years.

ACKNOWLEDGMENTS

We thank Nathanaël Fijalkow, Jürgen Giesl, Marcin Jurdcinski, Jan Křetínský, Orna Kupferman, and the anonymous reviewers for very helpful comments and remarks.

REFERENCES

[1] Tomás Babiak, Thomas Badie, Alexandre Duret-Lutz, Mojmír Kretínský, and Jan Strejcek. 2013. Compositional approach to suspension and other improvements to LTL translation. In *Model Checking Software*. Lecture Notes in Computer Science, Vol. 7976. Springer, 81–98. https://doi.org/10.1007/978-3-642-39176-7_6

[2] Udi Boker, Karoliina Lehtinen, and Salomon Sickert. 2022. On the translation of automata to linear temporal logic. In *Foundations of Software Science and Computation Structures*. Lecture Notes in Computer Science, Vol. 13242. Springer, 140–160. https://doi.org/10.1007/978-3-030-99253-8_8

[3] J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 6, 1-6 (1960), 66–92.

[4] Antonio Casares, Thomas Colcombet, and Nathanaël Fijalkow. 2021. Optimal transformations of games and automata using muller conditions. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Leibniz International Proceedings in Informatics, Vol. 198. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Article 123, 14 pages.

[5] Antonio Casares, Alexandre Duret-Lutz, Klara J. Meyer, Florian Renkin, and Salomon Sickert. 2022. Practical applications of the alternating cycle decomposition. In *Tools and Algorithms for the Construction and Analysis of Systems*. Lecture Notes in Computer Science, Vol. 13244. Springer, 99–117. https://doi.org/10.1007/978-3-030-99527-0_6

[6] Ivana Cerná and Radek Pelánek. 2003. Relating hierarchy of temporal properties to model checking. In *Mathematical Foundations of Computer Science 2003*. Lecture Notes in Computer Science, Vol. 2747. Springer, 318–327. https://doi.org/10.1007/978-3-540-45138-9_26

[7] Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer. 1981. Alternation. *J. ACM* 28, 1 (1981), 114–133. https://doi.org/10.1145/322234.322243

[8] Edward Y. Chang, Zohar Manna, and Amir Pnueli. 1992. Characterization of temporal property classes. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, Vol. 623. Springer, 474–486. https://doi.org/10.1007/3-540-55719-9_97

[9] Yaacov Choueka. 1974. Theories of automata on omega-tapes: A simplified approach. *J. Comput. Syst. Sci.* 8, 2 (1974), 117–141. https://doi.org/10.1016/S0022-0000(74)80051-6

[10] Volker Diekert and Paul Gastin. 2008. First-order definable languages. In *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*. Texts in Logic and Games, Vol. 2. Amsterdam University Press, 261–306.

[11] Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, and Henrich Lauko. 2022. From Spot 2.0 to Spot 2.10: What's new? In *Computer Aided Verification*. Lecture Notes in Computer Science, Vol. 13372. Springer, 174–187. https://doi.org/10.1007/978-3-031-13188-2_9

[12] Javier Esparza, Jan Kretínský, and Salomon Sickert. 2020. A unified translation of linear temporal logic to ω-automata. *J. ACM* 67, 6 (2020), Article 33, 61 pages.

[13] Javier Esparza, Rubén Rubio, and Salomon Sickert. 2022. A simple rewrite system for the normalization of linear temporal logic. In *Principles of Systems Design*. Lecture Notes in Computer Science, Vol. 13660. Springer, 208–227.

[14] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. 1997. First-order logic with two variables and unary temporal logic. In *LICS*. IEEE Computer Society, 228–235. https://doi.org/10.1109/LICS.1997.614950

[15] Dov M. Gabbay. 1987. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*. Lecture Notes in Computer Science, Vol. 398. Springer, 409–448. https://doi.org/10.1007/3-540-51803-7_36

[16] Dov M. Gabbay, Ian Hodkinson, and Mark A. Reynolds. 1994. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press on Demand.

[17] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. 1980. On the temporal analysis of fairness. In *POPL*. ACM Press, 163–173. https://doi.org/10.1145/567446.567462

[18] Dimitar P. Guelev. 2008. A syntactical proof of the canonical reactivity form for past linear temporal logic. *J. Log. Comput.* 18, 4 (2008), 615–623. https://doi.org/10.1093/logcom/exn002

[19] Sankar Gurumurthy, Orna Kupferman, Fabio Somenzi, and Moshe Y. Vardi. 2003. On complementing nondeterministic Büchi automata. In *Correct Hardware Design and Verification Methods*. Lecture Notes in Computer Science, Vol. 2860. Springer, 96–110. https://doi.org/10.1007/978-3-540-39724-3_10

[20] Tobias John, Simon Jantsch, Christel Baier, and Sascha Klüppelholz. 2021. Determinization and limit-determinization of Emerson-Lei automata. In *Automated Technology for Verification and Analysis*. Lecture Notes in Computer Science, Vol. 12971. Springer, 15–31. https://doi.org/10.1007/978-3-030-88885-5_2

[21] Tobias John, Simon Jantsch, Christel Baier, and Sascha Klüppelholz. 2022. From Emerson-Lei automata to deterministic, limit-deterministic or good-for-MDP automata. *Innov. Syst. Softw. Eng.* 18, 3 (2022), 385–403. https://doi.org/10.1007/s11334-022-00445-7

[22] Jan Kretínský, Tobias Meggendorfer, and Salomon Sickert. 2018. Owl: A library for ω-words, automata, and LTL. In *Automated Technology for Verification and Analysis*. Lecture Notes in Computer Science, Vol. 11138. Springer, 543–550. https://doi.org/10.1007/978-3-030-01090-4_34

[23] Orna Kupferman and Moshe Y. Vardi. 2001. Weak alternating automata are not that weak. *ACM Trans. Comput. Log.* 2, 3 (2001), 408–429. https://doi.org/10.1145/377978.377993

[24] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck. 1985. The glory of the past. In *Logic of Programs*. Lecture Notes in Computer Science, Vol. 193. Springer, 196–218. https://doi.org/10.1007/3-540-15648-8_16

[25] Christof Löding and Wolfgang Thomas. 2000. Alternating automata and logics over infinite words. In *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*. Lecture Notes in Computer Science, Vol. 1872. Springer, 521–535. https://doi.org/10.1007/3-540-44929-9_36

[26] Oded Maler. 2010. On the Krohn-Rhodes cascaded decomposition theorem. In *Time for Verification: Essays in Memory of Amir Pnueli*. Lecture Notes in Computer Science, Vol. 6200. Springer, 260–278. https://doi.org/10.1007/978-3-642-13754-9_12

[27] Oded Maler and Amir Pnueli. 1990. Tight bounds on the complexity of cascaded decomposition of automata. In *Proceedings of FOCS*. 672–682. https://doi.org/10.1109/FSCS.1990.89589

[28] Oded Maler and Amir Pnueli. 1994. On the cascaded decomposition of automata, its complexity and its application to logic. Unpublished. Retrieved March 14, 2024 from http://www-verimag.imag.fr/~maler/Papers/decomp.pdf

[29] Zohar Manna and Amir Pnueli. 1990. A hierarchy of temporal properties. In *PODC*. ACM, 377–410. https://doi.org/10.1145/93385.93442

[30] Zohar Manna and Amir Pnueli. 1991. Completing the temporal picture. *Theor. Comput. Sci.* 83, 1 (1991), 91–130. https://doi.org/10.1016/0304-3975(91)90041-Y

[31] Zohar Manna and Amir Pnueli. 1992. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, https://doi.org/10.1007/978-1-4612-0931-7

[32] Robert McNaughton and Seymour A. Papert. 1971. *Counter-Free Automata*. MIT Research Monograph No. 65). MIT Press.

[33] Philipp J. Meyer, Salomon Sickert, and Michael Luttenberger. 2018. Strix: Explicit reactive synthesis strikes back! In *Computer Aided Verification*. Lecture Notes in Computer Science, Vol. 10981. Springer, 578–586. https://doi.org/10.1007/978-3-319-96145-3_31

[34] Satoru Miyano and Takeshi Hayashi. 1984. Alternating finite automata on omega-words. *Theor. Comput. Sci.* 32 (1984), 321–330. https://doi.org/10.1016/0304-3975(84)90049-5

[35] David E. Muller, Ahmed Saoudi, and Paul E. Schupp. 1986. Alternating automata, the weak monadic theory of the tree, and its complexity. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, Vol. 226. Springer, 275–283. https://doi.org/10.1007/3-540-16761-7_77

[36] David E. Muller, Ahmed Saoudi, and Paul E. Schupp. 1988. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *LICS*. IEEE Computer Society, 422–427. https://doi.org/10.1109/LICS.1988.5139

[37] Daniel Oliveira and João Rasga. 2020. Revisiting separation: Algorithms and complexity. *Logic Journal of the IGPL* 29, 3 (2020), 251–302. https://doi.org/10.1093/jigpal/jzz081

[38] Radek Pelánek and Jan Strejcek. 2005. Deeper connections between LTL and alternating automata. In *Implementation and Application of Automata*. Lecture Notes in Computer Science, Vol. 3845. Springer, 238–249. https://doi.org/10.1007/11605157_20

[39] Nir Piterman and Amir Pnueli. 2018. Temporal logic and fair discrete systems. In *Handbook of Model Checking*. Springer, 27–73. https://doi.org/10.1007/978-3-319-10575-8_2

[40] Amir Pnueli. 1977. The temporal logic of programs. In *FOCS*. IEEE Computer Society, 46–57. https://doi.org/10.1109/SFCS.1977.32

[41] Amir Pnueli. 1981. The temporal semantics of concurrent programs. *Theor. Comput. Sci.* 13 (1981), 45–60. https://doi.org/10.1016/0304-3975(81)90110-9

[42] Mark Reynolds. 2000. More past glories. In *LICS*. IEEE Computer Society, 229–240. https://doi.org/10.1109/LICS.2000.855772

[43] Shmuel Safra. 1988. On the complexity of omega-automata. In *FOCS*. IEEE Computer Society, 319–327. https://doi.org/10.1109/SFCS.1988.21948

[44] Klaus Schneider. 2004. *Verification of Reactive Systems: Formal Methods and Algorithms*. Springer. https://doi.org/10.1007/978-3-662-10778-2

[45] Salomon Sickert and Javier Esparza. 2020. An efficient normalization procedure for linear temporal logic and very weak alternating automata. In *LICS*. ACM, 831–844. https://doi.org/10.1145/3373718.3394743

[46] Wolfgang Thomas. 1981. A combinatorial approach to the theory of omega-automata. *Inf. Control.* 48, 3 (1981), 261–283. https://doi.org/10.1016/S0019-9958(81)90663-X

[47] Moshe Y. Vardi. 1994. Nontraditional applications of automata theory. In *Theoretical Aspects of Computer Software*. Lecture Notes in Computer Science, Vol. 789. Springer, 575–597. https://doi.org/10.1007/3-540-57887-0_116

[48] Moshe Y. Vardi. 1996. An automata-theoretic approach to linear temporal logic. *Logics for Concurrency* 226 (1996), 238–266. https://doi.org/10.1007/3-540-60915-6_6

[49] Lenore D. Zuck. 1986. *Past Temporal Logic*. Ph.D. Dissertation. Weizmann Institute of Science, Israel.

Received 27 January 2023; revised 27 February 2024; accepted 1 March 2024