TEMPORAL LOGIC AND SEMIDIRECT PRODUCTS: AN EFFECTIVE CHARACTERIZATION OF THE UNTIL HIERARCHY

DENIS THÉRIEN †§ AND THOMAS WILKE ‡¶

Abstract. We reveal an intimate connection between semidirect products of finite semigroups and substitution of formulas in linear temporal logic. We use this connection to obtain an algebraic characterization of the until hierarchy of linear temporal logic. (The k-th level of that hierarchy is comprised of all temporal properties that are expressible by a formula of nesting depth k in the until operator.) Applying deep results from finite semigroup theory we are able to prove that each level of the until hierarchy is decidable. By means of Ehrenfeucht–Fraïssé games, we extend the results from linear temporal logic over finite sequences to linear temporal logic over infinite sequences.

Key words. linear temporal logic, until hierarchy, substitution, finite semigroups, pseudovarieties of semigroups, aperiodic semigroups, semidirect products

AMS subject classifications. 03B44, 03B70, 03D05, 68Q45, 68Q70, 20M20, 20M35

1. Introduction. Linear temporal logic is a language for expressing relationships between the order of events occurring over time. It is used in various areas of computer science as different as computer-aided verification (temporal logic as a specification language for reactive and concurrent systems) and databases (temporal logic to represent knowledge involving time and to deduct on time-dependent data). Its basic operators (modalities) are "next", "eventually", and "until", of which "until" is the most powerful one.[1] The latter observation has lead to a classification of temporal properties according to the nesting depth in the until operator required to express them. In [8], it was shown that the resulting hierarchy, the so-called "until hierarchy", is strict. In fact, "counting up to k" requires nesting depth k in the until operator (and nesting depth k is enough to express this property).

In this paper we show that the "until depth" of a given property (i. e., the lowest level of the until hierarchy the property belongs to) is effectively computable, both in the case where temporal logic is interpreted in finite sequences and in the case where it is interpreted in infinite sequences. This answers a question left open in [6] and [8].

Our approach to the computational problem of determining the until depth of a given property of finite sequences follows a well-known and established pattern. In a first step, we define a hierarchy of finite semigroups and prove that this hierarchy is tightly connected with the until hierarchy: a temporal property belongs to the k-th level of the until hierarchy if and only if its syntactic semigroup (the transformation semigroup of its minimal automaton) belongs to the k-th level of the hierarchy of semigroups. In a second step, we show that for a finite semigroup one can effectively compute the lowest level of the hierarchy of semigroups it belongs to. This yields the desired result, as the syntactic semigroup of a given temporal property can be computed effectively.

†School of Computer Science, McGill University, 3480 University Montréal, Québec H3A 2A7, Canada, denis@opus.cs.mcgill.ca.

‡Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany, wilke@ti.informatik.uni-kiel.de.

§Author is supported by FCAR and NSERC.

¶Part of this research was carried out while the second author was a post-doctoral fellow at DIMACS during the special year on Logic and Algorithms. This paper was written during the second author's stay at Institut Gaspard Monge, Université de Marne-la-Vallée.

[1] If one chooses the right variant of the until operator, the other two operators become superfluous, i. e., they are no longer needed to achieve the full power of temporal logic.

It is in the first step of the above process where we introduce new methodology. We develop a completely new approach to understanding semidirect products. We prove that semidirect products can be characterized in terms of substitution of formulas of linear temporal logic. Under reasonably weak assumptions, the semidirect product of two classes of semigroups corresponds to the set of formulas that are obtained by substituting formulas corresponding to the second class for atomic formulas in formulas corresponding to the first class. This is what we call the 'semidirect product/substitution principle'.

In the second step of the above process, we apply results from [18] and an intricate decidability criterion from [22] involving graph congruences to prove that each level of the hierarchy of semigroups in question is decidable.

The extension to the case where temporal logic is interpreted in infinite sequences is achieved by means of the Ehrenfeucht–Fraïssé game developed in [8]. This game gives us a way to approach the until hierarchy over infinite sequences from an algebraic point of view. We prove that the situation is exactly as in the case of finite sequences: a temporal property of infinite sequences belongs to the k-th level of the until hierarchy if and only if its syntactic semigroup (in the sense of [2]) belongs to the k-th level of the same semigroup hierarchy as above. The decidability result is then merely a corollary.

Temporal logic, first-order logic over unary predicates with total ordering, and star-free expressions are known to have the same expressive power, on both finite sequences and on infinite sequences [15, 9, 11, 10, 19]. Each of the mutually equivalent formalisms motivates its own hierarchy: the until hierarchy, the quantifier-alternation hierarchy, and the dot-depth hierarchy [5]. The two latter are strict [3] and identical [20], whereas [8] describes a family of properties that separates the levels of the until hierarchy but is contained in the second level of the quantifier-alternation/dot-depth hierarchy. It is a long standing problem whether the "dot depth" of a star-free language is computable.

Star-free languages, and hence the class of all temporal properties, have also been shown to correspond to aperiodic (group-free) semigroups ([15, 13]). The semigroup-theoretic characterization of the until hierarchy we give in this paper has a natural interpretation as a hierarchy for aperiodic semigroups, which is closely related to the Krohn–Rhodes Decomposition Theorem. This theorem is the most powerful tool for decomposing finite semigroups (respectively finite automata).

In [4] it is shown that level 0 of the until hierarchy over finite sequences is decidable. In [6] the hierarchy of semigroups we consider is defined and partial results about its relation to the until hierarchy over finite sequences are obtained (for more details, see the remark right after Theorem 4.7).

In § 2, we provide background on temporal logic and recall the definition of the until hierarchy over finite sequences. In § 3, we establish the connection between semidirect products and substitution. In § 4, we characterize the until hierarchy over finite sequences algebraically and show how the desired decidability result follows. In § 5, our results are extended to infinite sequences. Section 6 contains open problems.

This is a completely revised version of our technical report [23] and our conference paper [24].

For finite semigroup theory, the reader is referred to the excellent monograph [1], where our notation also stems from. The book [14] provides background on the algebraic theory of regular languages. The handbook chapters [7] and [21] survey temporal logic and the theory of regular ω-languages.

2. Future Temporal Logic on Strings and the Until Hierarchy. A *future temporal logic (FTL) formula* over a finite set P of propositional variables is built from the elements of P and the logical constant TRUE using the boolean connectives

$\neg$ and $\vee$ and the usual temporal logic operators: $\bigcirc$ (next), $\Diamond$ (eventually), and U (until), the two former of which are unary operators and the latter of which is a binary operator written in infix notation.

FTL formulas over a set P of propositional variables are interpreted in strings over an alphabet denoted Σ_P, which consists of all subsets of P. Given a string u, its length is denoted by $|u|$, and its positions are numbered $0, 1, 2, \ldots, |u|-1$. The symbol at position i is denoted $u(i)$.

Given a string $u \in \Sigma_P^+$ and a position $i < |u|$, one defines what it means for an FTL formula φ over P to be true in u at i, in symbols $(u, i) \models \varphi$. In this paper, we adopt the following conventions:

- $(u, i) \models \textsc{True}$,
- $(u, i) \models p$ if $p \in u(i)$,
- $(u, i) \models \neg\varphi$ if not $(u, i) \models \varphi$,
- $(u, i) \models \varphi_1 \vee \varphi_2$ if $(u, i) \models \varphi_1$ or $(u, i) \models \varphi_2$,
- $(u, i) \models \bigcirc\varphi$ if $i + 1 < |u|$ and $(u, i + 1) \models \varphi$,
- $(u, i) \models \Diamond\varphi$ if there exists j with $i \leq j < |u|$ such that $(u, j) \models \varphi$,
- $(u, i) \models \varphi_1 \mathbin{\mathsf{U}} \varphi_2$ if there exists j with $i \leq j < |u|$ such that $(u, i') \models \varphi_1$ for every i' with $i \leq i' < j$ and $(u, j) \models \varphi_2$.

Given an FTL formula φ, we write $u \models \varphi$ and say that u is a model of φ if $(u, 0) \models \varphi$. We write $L(\varphi)$ for the set of all models of φ, i. e., $L(\varphi) = \{u \in \Sigma_P^+ \mid u \models \varphi\}$. We say that φ expresses $L(\varphi)$.

Similarly, we say that a language L is expressible in FTL if there exists an FTL formula φ such that $L = L(\varphi)$. Given a class Φ of FTL formulas, we write $L(\Phi)$ for $\{L(\varphi) \mid \varphi \in \Phi\}$.

In some of the commonly accepted definitions of the semantics of the eventually operator the value of j is restricted to positions strictly larger than i. For the until operator, a similar restriction of i' to the values strictly between i and j has been considered. The until depth of a language, however, does not depend on which of the four possible combinations is actually chosen. (For a formal definition of until depth, see below.)

The *until depth* of an FTL formula is its nesting depth in terms of the until operator. When φ is a formula, $\mathrm{ud}(\varphi)$ denotes its until depth. Formally, the until depth is defined inductively by:

$$\begin{aligned}
\mathrm{ud}(\textsc{True}) &= 0, \\
\mathrm{ud}(p) &= 0, \qquad \text{for } p \in P, \\
\mathrm{ud}(\neg\varphi) &= \mathrm{ud}(\varphi)\ , \\
\mathrm{ud}(\varphi_1 \vee \varphi_2) &= \max\{\mathrm{ud}(\varphi_1), \mathrm{ud}(\varphi_2)\}\ , \\
\mathrm{ud}(\bigcirc\varphi) &= \mathrm{ud}(\varphi)\ , \\
\mathrm{ud}(\Diamond\varphi) &= \mathrm{ud}(\varphi)\ , \\
\mathrm{ud}(\varphi_1 \mathbin{\mathsf{U}} \varphi_2) &= \max\{\mathrm{ud}(\varphi_1), \mathrm{ud}(\varphi_2)\} + 1\ .
\end{aligned}$$

The class of all FTL formulas of until depth at most k is denoted by $\mathtt{Ut}_k$. That is, a language L belongs to $L(\mathtt{Ut}_k)$ if and only if it is expressed by some FTL formula of until depth at most k. The until hierarchy is the sequence $L(\mathtt{Ut}_0)$, $L(\mathtt{Ut}_1)$, $L(\mathtt{Ut}_2)$, ... In [8], it is shown that this hierarchy is strict in the following sense: $L(\mathtt{Ut}_0) \subsetneq L(\mathtt{Ut}_1) \subsetneq L(\mathtt{Ut}_2) \subsetneq \cdots$.

The main result of this paper is Theorem 4.18, which states that given an FTL formula φ or a regular language L expressible in FTL, one can compute the lowest level of the until hierarchy $L(\varphi)$, respectively L, belongs to, i. e., the smallest k such that $L(\varphi) \in L(\mathtt{Ut}_k)$, respectively $L \in L(\mathtt{Ut}_k)$.— Theorem 5.9 states the same result for ω-words instead of strings.

3. Substitution and Semidirect Products. This section builds the core of the paper. It provides the correspondence between semidirect products of pseudovarieties of semigroups and substitution in past temporal logic formulas.

3.1. Past temporal logic. *Past temporal logic (PTL) formulas* are built exactly as FTL formulas but use a corresponding set of reversed operators: instead of $\bigcirc$, $\Diamond$, and U, PTL formulas use $\ominus$ (previously), $\diamondminus$ (sometime in the past, once upon a time), and S (since). The semantics of PTL formulas is defined analogously to the semantics of FTL formulas: in the above definition for FTL, every temporal operator is replaced by its reversed form and every condition that involves the ordering is replaced by the symmetric condition:

- $(u,i) \models \ominus\varphi$ if $i > 0$ and $(u, i-1) \models \varphi$,
- $(u,i) \models \diamondminus\varphi$ if there exists j with $j \leq i$ such that $(u,j) \models \varphi$,
- $(u,i) \models \varphi_1 \mathsf{S} \varphi_2$ if there exists j with $j \leq i$ such that $(u,i') \models \varphi_1$ for every i' with $j < i' \leq i$ and $(u,j) \models \varphi_2$.

Given a PTL formula φ over a set P of propositional variables and $u \in \Sigma_P^+$, we write $u \models \varphi$ and say that u is a model of φ if $(u, |u|-1) \models \varphi$. As a consequence, if u is a string and v its reverse, and if φ is an FTL formula and ψ its reversed counterpart (obtained by replacing every operator by its reversal), then $u \models \varphi$ iff $v \models \psi$.

Abusing notation, we use $L(\varphi)$ to denote the set of models of both FTL and PTL formulas although there are formulas that are FTL and PTL formulas at the same time and their respective sets of models are distinct; consider, e. g., the atomic formula p. It will, however, always be clear from the context whether we view a given formula as an FTL or a PTL formula. Similarly, we use $L(\Phi)$ to also denote the class of all languages expressible by some PTL formula from Φ.

The *since depth* of a PTL formula is its nesting depth in terms of the since operator; we denote the class of all PTL formulas of since depth at most k by $\mathtt{Sn}_k$. It should be clear that a language belongs to $L(\mathtt{Ut}_k)$ if and only if its reverse belongs to $L(\mathtt{Sn}_k)$.

3.2. Definition of substitution. If φ is a PTL formula over a set of propositions P and $(\psi_p)_{p\in P}$ a family of PTL formulas over Q, then $\varphi[p \mapsto \psi_p]$ stands for the PTL formula over Q obtained from φ by simultaneously replacing every occurrence of a propositional variable $p \in P$ by ψ_p.

Formally, substitution is defined by:

$$\text{TRUE}[p \mapsto \psi_p] = \text{TRUE} \ , \tag{3.1}$$

$$p'[p \mapsto \psi_p] = \psi_{p'} \ , \qquad \text{for } p' \in P \ , \tag{3.2}$$

$$\neg\varphi[p \mapsto \psi_p] = \neg(\varphi[p \mapsto \psi_p]) \ , \tag{3.3}$$

$$\varphi_1 \vee \varphi_2[p \mapsto \psi_p] = (\varphi_1[p \mapsto \psi_p]) \vee (\varphi_2[p \mapsto \psi_p]) \ , \tag{3.4}$$

$$\ominus\varphi[p \mapsto \psi_p] = \ominus(\varphi[p \mapsto \psi_p]) \ , \tag{3.5}$$

$$\diamondminus\varphi[p \mapsto \psi_p] = \diamondminus(\varphi[p \mapsto \psi_p]) \ , \tag{3.6}$$

$$\varphi_1 \mathsf{S} \varphi_2[p \mapsto \psi_p] = (\varphi_1[p \mapsto \psi_p]) \mathsf{S} (\varphi_2[p \mapsto \psi_p]) \ . \tag{3.7}$$

3.3. Basic properties of substitution. Obviously, substitution is associative:

REMARK 1 (associativity of substitution). *Let φ be a PTL formula over P, $\{\psi_p\}_{p\in P}$ a family of PTL formulas over Q, and $\{\chi_q\}_{q\in Q}$ a family of PTL formulas over R.*

Then

$$\varphi[p \mapsto \psi_p[q \mapsto \chi_q]] = \varphi[p \mapsto \psi_p][q \mapsto \chi_q] \ . \tag{3.8}$$

In first-order logic, the most important property of substitution is described by the so-called substitution lemma. There is an analogue with temporal logic. For a family $(\psi_p)_{p\in P}$ of PTL formulas over Q and a string $u \in \Sigma_Q^+$, define a string over Σ_P, denoted $u[p \mapsto \psi_p]$, as follows: $u[p \mapsto \psi_p] = v_0 \dots v_{|u|-1}$ where $v_i = \{p \in P \mid (u,i) \models \psi_p\}$. Using this notation, the substitution lemma for temporal logic can be stated as follows.

LEMMA 3.1 (substitution lemma for PTL). *Let φ be a PTL formula over P and $(\psi_p)_{p\in P}$ a family of PTL formulas over Q.*

For every string u over Σ_Q and every position $i < |u|$, the following then holds:

$$(u,i) \models \varphi[p \mapsto \psi_p] \qquad \textit{iff} \qquad (u[p \mapsto \psi_p], i) \models \varphi \ . \tag{3.9}$$

This can be proven by a straightforward induction on the structure of φ.

An immediate consequence of the substitution lemma is:

COROLLARY 3.2. *Let φ and φ' be PTL formulas over P, and $\{\psi_p\}_{p\in P}$ and $\{\psi'_p\}_{p\in P}$ families of PTL formulas over Q such that $L(\varphi) = L(\varphi')$ and $L(\psi_p) = L(\psi'_p)$ for every $p \in P$.*

Then $L(\varphi[p \mapsto \psi_p]) = L(\varphi'[p \mapsto \psi'_p])$.

3.4. Substitution and sequential functions. Recall that a function $\sigma\colon A^+ \to B^+$ is called a sequential function if it is realized by a transducer. A transducer is a tuple

$$(\mathcal{Q}, A, B, s_I, \delta, \lambda) \tag{3.10}$$

where $\mathcal{Q}$ is a finite set of states, $q_I \in \mathcal{Q}$ is the initial state, A and B are alphabets, $\delta\colon \mathcal{Q}\times A \to \mathcal{Q}$ is a transition function, and $\lambda\colon \mathcal{Q}\times A \to B^+$ is an output function. The transition function δ is inductively extended to a function $\mathcal{Q} \times A^+ \to \mathcal{Q}$ by setting $\delta(q, wa) = \delta(\delta(q,w), a)$. The function $\sigma\colon A^+ \to B^+$ realized by the transducer is inductively defined by

$$\begin{array}{rcll} \sigma(a) & = & \lambda(q_I, a) & \text{for } a \in A, \\ \sigma(wa) & = & \sigma(w)\, \lambda(\delta(q_I, w), a) & \text{for } a \in A,\ w \in A^+. \end{array}$$

A transducer as in (3.10) is called synchronous if $\lambda(q,a) \in B$ for all $q \in \mathcal{Q}$, $a \in A$. Accordingly, a sequential function is called synchronous if it is realized by a synchronous transducer.

Under a certain proviso inverse images of synchronous sequential functions can be described by substitution:

LEMMA 3.3. *Let $\sigma\colon \Sigma_Q^+ \to \Sigma_P^+$ be a synchronous sequential function and $\{\psi_p\}_{p\in P}$ a family of PTL formulas over Q such that*

$$\sigma^{-1}(L(p)) = L(\psi_p) \tag{3.11}$$

for every $p \in P$.

Then, for every PTL formula φ over P,

$$\sigma^{-1}(L(\varphi)) = L(\varphi[p \mapsto \psi_p]) \ . \tag{3.12}$$

Proof. The proof is by induction on the structure of φ. The base case is trivial. In the induction step, we have to consider boolean connectives and temporal operators. Boolean connectives are dealt with easily, for inverse functions respect boolean operations. Of the three temporal operators, we only consider since, which requires the most involved argument. Given a string u and a position $i < |u|$, we will use $u[0,i]$ to denote the prefix $u(0)\dots u(i)$ of u. Observe that, in general, $u[0,i] \models \varphi$ is equivalent to $(u,i) \models \varphi$ for every PTL formula φ.

Assume φ is of the form $\varphi_1 \mathsf{S} \varphi_2$, and let u be an arbitrary string over Σ_Q. Let $v = \sigma(u)$. Then, since σ is assumed to be synchronous, $|v| = |u|$. The string u belongs to $\sigma^{-1}(L(\varphi_1 \mathsf{S} \varphi_2))$ if and only if there exists $i < |v|$ such that

- $v[0,i] \models \varphi_2$, and
- $v[0,j] \models \varphi_1$ for every j with $i < j < |v|$.

By induction hypothesis, we know that $v[0,i] \models \varphi_2$ if and only if $u[0,i] \models \varphi_2[p \mapsto \psi_p]$. (Observe that, since σ is synchronous, $\sigma(u[0,i]) = v[0,i]$.) Similarly, $v[0,j] \models \varphi_1$ if and only if $u[0,j] \models \varphi_1[p \mapsto \psi_p]$ for every $j < |u|$. Therefore, the above assertion is equivalent to the existence of $i \leq |u|$ such that

- $u[0,i] \models \varphi_2[p \mapsto \psi_p]$, and
- $u[0,j] \models \varphi_1[p \mapsto \psi_p]$ for every j with $i < j$.

According to the semantics of PTL formulas, the latter is true if and only if $u \models (\varphi_1[p \mapsto \psi_p]) \mathsf{S} (\varphi_2[p \mapsto \psi_p])$. However, in view of (3.7), $(\varphi_1[p \mapsto \psi_p]) \mathsf{S} (\varphi_2[p \mapsto \psi_p])$ is the same as $(\varphi_1 \mathsf{S} \varphi_2)[p \mapsto \psi_p]$. ∎

The previous lemma actually holds under seemingly weaker assumptions; beside (3.11) we only need that σ is *length-preserving,* i. e., that σ satisfies $|\sigma(u)| = |u|$ for every $u \in \Sigma_Q^+$, and that it *respects prefixes* in the following sense: $\sigma(uv) \in \sigma(u)\Sigma_P^+$ for all $u, v \in \Sigma_Q^+$. It is, however, easy to see that if these conditions are met, then σ is a synchronous sequential function.

3.5. Substitution and homomorphisms. One way to interpret Lemma 3.3 is to say that substitution can (in many cases) be regarded as applying the inverse of a sequential function. If the substituents are very simple, substitution can even be explained using homomorphisms.

LEMMA 3.4. *Let φ be a PTL formula over P and $\{\psi_p\}_{p \in P}$ a P-indexed family of PTL formulas over Q. Furthermore, assume each formula ψ_p is a disjunction of propositional variables from Q.*

Then $L(\varphi[p \mapsto \psi_p])$ is the preimage of $L(\varphi)$ under a homomorphism $\Sigma_Q^+ \to \Sigma_P^+$.

By convention, an empty disjunction is FALSE.

Proof. For every $p \in P$, let D_p denote the set of disjuncts of ψ_p. Consider the homomorphism $g\colon \Sigma_Q^+ \to \Sigma_P^+$ defined by $g(b) = \bigcup\{p \mid D_p \cap b \neq \emptyset\}$. Let $v \in \Sigma_Q^*$ and $b \in \Sigma_Q$, and assume $g(vb) = ua$ where $a \in \Sigma_P$. Then $p \in a$ if and only if there exists $q \in b$ such that $q \in D_p$. The latter is equivalent to $vb \models \psi_p$. Thus, $g^{-1}(L(p)) = L(\psi_p)$, and we can apply Lemma 3.3 to conclude $g^{-1}(L(\varphi)) = L(\varphi[p \mapsto \psi_p])$. ∎

This lemma can be strengthened:

LEMMA 3.5. *Let φ be a PTL formula over P and $\{\psi_p\}_{p \in P}$ a P-indexed family of PTL formulas over Q. Furthermore, assume each formula ψ_p is a boolean combination of propositional variables from Q (i. e., a propositional formula).*

Then $L(\varphi[p \mapsto \psi_p])$ is the preimage of $L(\varphi)$ under a homomorphism $\Sigma_Q^+ \to \Sigma_P^+$.

Proof. For $b \in \Sigma_Q$, let α_b be a propositional formula such that $L(\alpha_b) = \Sigma_Q^* b$. For each $p \in P$, define a formula μ_p over Σ_Q (!) as follows. Let μ_p be the disjunction of all $b \in \Sigma_Q$ such that α_b implies ψ_p (as a propositional formula). Then, $L(\psi_p) = L(\mu_p[b \mapsto \alpha_b])$, and thus $L(\varphi[p \mapsto \psi_p]) = L(\varphi[p \mapsto \mu_p][b \mapsto \alpha_b])$. Write φ' for $\varphi[p \mapsto \mu_p]$. By the previous lemma, $L(\varphi') = g^{-1}(L(\varphi))$ for some homomorphism $g\colon \Sigma_{\Sigma_Q}^+ \to \Sigma_P^+$.

Consider now the homomorphism $h\colon \Sigma_Q^+ \to \Sigma_{\Sigma_Q}^+$ defined by $h(b) = \{b\}$. Trivially, $h^{-1}(L(b)) = L(\alpha_b)$. Lemma 3.3 then implies $L(\varphi'[b \mapsto \alpha_b]) = h^{-1}(L(\varphi'))$, which means $L(\varphi[p \mapsto \psi_p]) = h^{-1}(g^{-1}(L(\varphi)))$. ∎

3.6. Substitution on classes of formulas. Pseudovarieties of semigroups are classes of finite semigroups closed under homomorphic images, finite direct products, and subsemigroups, see [1, p. 53]. With each such class $\mathbf{V}$, we associate the class $L(\mathbf{V})$ of all languages over alphabets of the form Σ_P that are recognized by elements from $\mathbf{V}$. Recall that a language $L \subseteq A^+$ is said to be recognized by a semigroup S if there exists a semigroup homomorphism $h: A^+ \to S$ and a subset $F \subseteq S$ such that $L = h^{-1}(F)$, i. e., if L is the preimage of some subset of S under h.

Complex pseudovarieties of semigroups are often decomposed using semidirect products, that is, they are described as a semidirect product of simpler pseudovarieties. As usual, we denote the semidirect product of two pseudovarieties $\mathbf{V}$ and $\mathbf{W}$ by $\mathbf{V} * \mathbf{W}$, see [1, pp. 266, 269].

On classes of formulas, we now define an operation, denoted $\star$, which corresponds to the semidirect product on classes of semigroups, as we will see below in Theorem 3.6.

Given classes Φ and Ψ of PTL formulas, we define $\Phi \star \Psi$ to contain all formulas that are boolean combinations of formulas from Ψ and formulas that can be written in the form $\varphi[p \mapsto \psi_p]$ where $\varphi \in \Phi$ and the ψ_p's themselves are boolean combinations of formulas from Ψ and propositional variables.

Before we investigate the close connection between $\star$ and $*$, let's observe that $\star$ is a convenient tool for defining classes of formulas. For instance, using $\star$ the k-th level of the since hierarchy can be expressed in terms of its first level:

$$\mathtt{Sn}_k = \underbrace{\mathtt{Sn}_1 \star \cdots \star \mathtt{Sn}_1}_{k \text{ factors}} \qquad \text{for } k > 0, \tag{3.13}$$

Notice that $\star$ is associative. So grouping is not necessary in the above equation.

As is usual with associative binary operations, we allow ourselves to use exponentiation to denote "iterated multiplication", e. g., we allow ourselves to write $\mathtt{Sn}_1^k$ for the right hand side of the above equation.

3.7. The semidirect product/substitution principle. In this subsection we relate the two operations $\star$ and $*$ to each other. There are two parts to this relation, Propositions 3.8 and 3.10. Combined into one statement, they yield the following concise theorem. The set of all PTL formulas without occurrence of $\diamondminus$ and S is denoted by $\mathtt{Pv}$.

THEOREM 3.6 (semidirect product/substitution principle, SSP). *Let $\mathbf{V}$, $\mathbf{W}$ be pseudovarieties of semigroups and Φ, Ψ classes of PTL formulas such that $L(\mathbf{V}) = L(\Phi)$ and $L(\mathbf{W}) = L(\Psi)$.*

If $L(\Phi \star \mathtt{Pv} \star \Psi) \subseteq L(\Phi \star \Psi)$, then

$$L(\mathbf{V} * \mathbf{W}) = L(\Phi \star \Psi) \ . \tag{3.14}$$

The theorem follows from Propositions 3.8 and 3.10 below, each of which proves one inclusion of (3.14).

In the proof of Proposition 3.8 we will make use of the following fact about wreath products, see [1, pp. 265, 266], and regular languages, see, e. g., [14]. (Wreath products are closely related to semidirect products and come in handy whenever one works with transformation semigroups.) Recall that the transformation semigroup of a transducer as in (3.10) is (Q, T) where T is the subsemigroup of Q^Q generated by $q \mapsto \delta(q, a)$ for $a \in A$.

PROPOSITION 3.7. *Let $\sigma: A^+ \to B^+$ be a sequential function given by a transducer whose transformation semigroup is (Q, T), and let $L \subseteq B^+$ be a language recognized by a semigroup S.*

Then $\sigma^{-1}(L)$ is recognized by the semigroup of $(S^1, S) \circ (Q, T)$, which is isomorphic to a particular semidirect product $S^Q * T$.

The inclusion from right to left in (3.14) now follows from the following proposition.

PROPOSITION 3.8 (SSP I). *Let* $\mathbf{V}$, $\mathbf{W}$ *be pseudovarieties of semigroups and* Φ, Ψ *classes of PTL formulas such that* $L(\mathbf{V}) = L(\Phi)$ *and* $L(\mathbf{W}) = L(\Psi)$.

Then

$$L(\Phi \star \Psi) \subseteq L(\mathbf{V} * \mathbf{W}) \ . \tag{3.15}$$

Proof. Assume $\varphi \in \Phi$ is a formula over P and $\{\psi_p\}_{p\in P}$ is a family of formulas over Q where each ψ_p is a boolean combination of propositional variables and formulas from Ψ. We will show that $L(\varphi[p \mapsto \psi_p])$ is recognized by an element of $\mathbf{V} * \mathbf{W}$. This is enough for two reasons. First, $\mathbf{W} \subseteq \mathbf{V} * \mathbf{W}$ holds for arbitrary pseudovarieties of semigroups, unless $\mathbf{V}$ is the empty pseudovariety or the pseudovariety consisting of the empty semigroup only. This means $L(\mathbf{W}) \subseteq L(\mathbf{V} * \mathbf{W})$. Second, $L(\mathbf{V} * \mathbf{W})$ is closed under boolean combinations, as $L(\mathbf{U})$ is closed under boolean combination for any arbitrary pseudovariety $\mathbf{U}$ of semigroups.

Let $Q = \{q_0, \dots, q_{m-1}\}$ and assume $\psi_p = \alpha_p(q_0, \dots, q_{m-1}, \mu_0, \dots, \mu_{n-1})$ where α_p is a boolean combination and the μ_j's are from Ψ. Since $\mathbf{W}$ is assumed to be a pseudovariety, we can find a single semigroup $T \in \mathbf{W}$ and a homomorphism $h\colon \Sigma_Q^+ \to T$ such that every language $L(\mu_j)$ is recognized by h, i. e., for every $j < n$ there exists $F_j \subseteq T$ such that $L(\mu_j) = h^{-1}(F_j)$. For every $j < n$, let ν_j be the formula $\bigvee_{t\in F_j} t$. And for each $t \in T$, let κ_t be a formula over Q from Ψ such that $L(\kappa_t) = h^{-1}(t)$. Then

$$L(\psi_p) = L(\alpha_p(\bar{q}, \bar{\nu})[q/t \mapsto q/\kappa_t]) \ , \tag{3.16}$$

where $[q/t \mapsto q/\kappa_t]$ means that every element from Q is replaced by itself and every element from t is replaced by κ_t; overstriking indicates a sequence of formulas.

Write β_p for $\alpha_p(\bar{q}, \bar{\nu})$. Then

$$L(\varphi[p \mapsto \psi_p]) = L(\varphi[p \mapsto \beta_p][q/t \mapsto q/\kappa_t]) \ . \tag{3.17}$$

Denote $\varphi[p \mapsto \beta_p]$ by φ'. From Lemma 3.4, we can conclude that $L(\varphi')$ is recognized by an element of $\mathbf{V}$, say S, as $L(\mathbf{V})$ is closed under inverse images.

Consider the sequential function σ defined by the transducer

$$(T^1, \Sigma_Q, \Sigma_{T\cup Q}, 1, \delta, \lambda) \tag{3.18}$$

as specified through:

$$\delta(t, a) = t\, h(a) \ , \tag{3.19}$$

$$\lambda(t, a) = \{t\, h(a)\} \cup a \qquad \text{for } t \in T^1,\ a \in \Sigma_Q. \tag{3.20}$$

The transformation semigroup of this transducer is isomorphic to (T^1, T). We have $\sigma(ua) \in \Sigma_T^*(\{h(ua)\} \cup a)$ for $u \in \Sigma_Q^*$, $a \in \Sigma_Q$, i. e., $\sigma^{-1}(L(t)) = h^{-1}(t)$ for every $t \in T$, hence $\sigma^{-1}(L(t)) = L(\kappa_t)$ for every $t \in T$. Also, $\sigma^{-1}(L(q)) = L(q)$ for every $q \in Q$. Thus, by Lemma 3.3, $L(\varphi'[q/t \mapsto q/\kappa_t]) = \sigma^{-1}(L(\varphi'))$. On the other hand, $\sigma^{-1}(L(\varphi'))$ is recognized by the semigroup of $(S^1, S) \circ (T^1, T)$ according to Proposition 3.7. Hence, $L(\varphi'[q/t \mapsto q/\kappa_t])$ is recognized by $S^{S^1} * T$, which is an element of $\mathbf{V} * \mathbf{W}$. ◻

To prove the other inclusion of (3.14), we need the so-called wreath product principle, introduced in [17]. For the use with temporal logic, it can be formulated as follows.

PROPOSITION 3.9 (wreath product principle). *Let* $L \subseteq \Sigma_P^+$ *be a language recognized by the semigroup of a wreath product* $(\mathcal{P}, S) \circ (\mathcal{Q}, T)$ *of transformation* *semigroups via a homomorphism* $h\colon \Sigma_P^+ \to S^{\mathcal{Q}} * T$. *Denote the projection of* h *onto the second component by* h_2, *and let* $R = T^1 \cup P$. *Define* $\sigma\colon \Sigma_P^+ \to \Sigma_R^+$ *by* $\sigma(a_0 \dots a_{n-1}) = b_0 \dots b_{n-1}$ *where* $b_i = \{h_2(a_0 \dots a_{i-1})\} \cup a_i$. *(By convention,* $h(\epsilon) = 1$.*)*

This function is a synchronous sequential function, which can be realized by a transducer whose transformation semigroup is T*'s right regular representation.*

The language L *is a boolean combination of languages of the following two types:*

(i) *languages of the form* $h_2^{-1}(t)$ *for some* $t \in T$*;*
(ii) *languages of the form* $\sigma^{-1}(V)$ *where* $V \subseteq \Sigma_R^+$ *is recognized by* S.

The other inclusion of (3.14) now follows from:

PROPOSITION 3.10 (SSP II). *Let* $\mathbf{V}$, $\mathbf{W}$ *be pseudovarieties of semigroups and* Φ, Ψ *classes of PTL formulas such that* $L(\mathbf{V}) = L(\Phi)$ *and* $L(\mathbf{W}) = L(\Psi)$.

Then

$$L(\mathbf{V} * \mathbf{W}) \subseteq L(\Phi \star \mathtt{Pv} \star \Psi) \ . \tag{3.21}$$

Proof. Assume $L \subseteq \Sigma_P^+$ is a language recognized by an element of $\mathbf{V} * \mathbf{W}$. There exist transformation semigroups $(\mathcal{P}, S)$ and $(\mathcal{Q}, T)$ with $S \in \mathbf{V}$ and $T \in \mathbf{W}$ such that L is recognized by the semigroup of $(\mathcal{P}, S) \circ (\mathcal{Q}, T)$ via some homomorphism $h\colon \Sigma_P^+ \to S^{\mathcal{Q}} * T$.

Let R, h_2, and σ be as in Proposition 3.9. It is enough to show that the two types of languages described in Proposition 3.9 can be defined by formulas from $\Phi \star \mathtt{Pv} \star \Psi$.

Clearly, every language of type (i) is definable by a formula from Ψ. To complete the proof, we show that every language of type (ii) is definable by a formula from $\Phi \star \mathtt{Pv} \star \Psi$.

Let $V \subseteq \Sigma_R^+$ be a language recognized by S. By assumption, there exists a formula $\varphi \in \Phi$ such that $V = L(\varphi)$. Also, for every $t \in T$, there exists a formula $\mu_t \in \Psi$ such that $h_2^{-1}(t) = L(\mu_t)$.

We want to apply Lemma 3.3. For every $r \in R$, we construct a formula ψ_r such that $\sigma^{-1}(L(r)) = L(\psi_r)$. By definition of σ, the ψ_r's can be chosen as follows:

$$\psi_r = \begin{cases} r & \text{if } r \in \mathcal{P}, \\ \ominus\mu_r & \text{if } r \in T \setminus \{1\}, \\ \ominus\mu_1 \wedge \neg\ominus\textsc{True} & \text{if } r = 1 \text{ and } 1 \in T, \\ \neg\ominus\textsc{True} & \text{if } r = 1 \text{ and } 1 \notin T. \end{cases}$$

(Notice that $\neg\ominus\textsc{True}$ defines the set of all strings of length exactly 1.) We now can apply Lemma 3.3 and conclude $\sigma^{-1}(L(\varphi)) = L(\varphi[r \mapsto \psi_r])$, hence $\sigma^{-1}(V) = L(\varphi[r \mapsto \psi_r])$. Obviously, every formula ψ_r belongs to $\mathtt{Pv} \star \Psi$ or else is atomic. □

4. Since and Until Hierarchy. In this section, we first obtain an algebraic characterization of the levels of the since hierarchy and will then be able to deduce that each level of this hierarchy is decidable. By the symmetry between FTL and PTL, this also implies that the until hierarchy is decidable.

There are essentially two ways of obtaining an effective characterization of the since hierarchy. In the first approach, one first shows that the k-th level of the since hierarchy is characterized by the left-hand side of (4.30), then shows (4.30), and finally proves that membership in the pseudovariety on the right-hand side of (4.30) is decidable. This approach was taken in our technical report [23] and relies heavily on fundamental results from [16]. The second approach is taken here. In it most of the algebraic arguments from [16] are replaced by syntactic/logical arguments. In that way, some of the results from [16] can actually be reproved, which is demonstrated in Subsection 4.4.

4.1. A normal form for PTL. We first prove a strong normal form for temporal logic which replaces the complicated algebraic argument of Subsection 6.1 of our technical report [23]. The normal form is established by syntactic means.

The observation that in PTL application of the previously operator can be restricted to propositional variables or negated propositional variables is folklore. It follows from the following (almost obvious) lemma.

LEMMA 4.1 (switching rules for $\ominus$). *For arbitrary PTL formulas φ and ψ,*

$$\begin{aligned} L(\ominus(\varphi_1 \mathsf{S}\, \varphi_2)) &= L(\ominus\varphi_1 \mathsf{S} \ominus\varphi_2) \ , \\ L(\ominus\diamond\varphi) &= L(\diamond\ominus\varphi) \ , \\ L(\ominus(\varphi \vee \psi)) &= L(\ominus\varphi \vee \ominus\psi) \ , \\ L(\ominus\neg\varphi) &= L(\neg\ominus\varphi \wedge \ominus\textsc{True}) \ . \end{aligned}$$

Writing $\mathtt{On}$ (reminding of "once upon a time") for the class of all PTL formulas not using S and $\ominus$, we obtain as an immediate consequence the following identity:

$$L(\mathtt{Sn}_0) = L(\mathtt{On} \star \mathtt{Pv}) \ . \tag{4.1}$$

We will extend this to any level of the since hierarchy. Let $\underline{\mathtt{Sn}}_k$ denote all formulas in $\mathtt{Sn}_k$ that don't use $\ominus$ or $\diamond$. Then:

THEOREM 4.2. *For every $k \geq 0$,*

$$L(\mathtt{Sn}_k) = L(\mathtt{On} \star \underline{\mathtt{Sn}}_k \star \mathtt{Pv}) \ . \tag{4.2}$$

In other words,

$$L(\mathtt{Sn}_k) = L(\mathtt{On} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv}) \qquad \text{for } k \geq 0. \tag{4.3}$$

Obviously, it is enough to prove (4.2) for $k = 1$. But this follows immediately from Lemma 4.1 and the following Lemma 4.3. The operator $\boxminus$ is defined by $\boxminus\varphi = \neg\diamond\neg\varphi$ and using it makes the formulas in Lemma 4.3 better readable.

LEMMA 4.3 (switching rules for $\diamond$ and S). *For arbitrary PTL formulas φ, ψ, and ρ,*

$$\begin{aligned} L((\varphi \wedge \psi) \mathsf{S}\, \rho) &= L((\varphi \mathsf{S}\, \rho) \wedge (\psi \mathsf{S}\, \rho)) \ , \\ L((\varphi \mathsf{S}\, (\psi \vee \rho)) &= L((\varphi \mathsf{S}\, \psi) \vee (\varphi \mathsf{S}\, \rho)) \ , \\ L((\varphi \vee \diamond q) \mathsf{S}\, \rho) &= L((\varphi \mathsf{S}\, \rho) \vee \diamond(\diamond q \wedge \ominus(\varphi \mathsf{S}\, \rho))) \ , \\ L((\varphi \vee \boxminus\psi) \mathsf{S}\, \rho) &= L((\varphi \mathsf{S}\, \rho) \vee (\diamond\rho \wedge \boxminus(\varphi \vee \boxminus\psi))) \ , \\ L(\varphi \mathsf{S}\, (\psi \wedge \diamond\rho)) &= L((\varphi \mathsf{S}\, \psi) \wedge \diamond(\psi \wedge \diamond\rho)) \ , \\ L(\varphi \mathsf{S}\, (\psi \wedge \boxminus\rho)) &= L(\varphi \mathsf{S}\, (\psi \wedge \rho) \wedge \boxminus(\boxminus\rho \vee (\varphi \wedge \varphi \mathsf{S}\, (\psi \wedge \rho)))) \ . \end{aligned}$$

Proof. We prove only the inclusion from right to left in the last equation, which is the only non-trivial assertion. Write φ_L and φ_R for the left respectively right hand side of this equation.

Assume $u \models \varphi_R$. Then there exists a position i in u such that

$$(u, i) \models \psi \ , \tag{4.4}$$

$$(u, i) \models \rho \ , \tag{4.5}$$

$$(u, j) \models \varphi \qquad \text{for every } j \text{ with } i < j < |u|. \tag{4.6}$$

Fix i to be the minimal such position. Clearly, if $(u,i) \models \boxminus \rho$, then $u \models \varphi_L$. So, for the rest, we assume

$$(u,i) \not\models \boxminus \rho \tag{4.7}$$

and show this assumption leads to a contradiction. As the second conjunct of φ_R holds at i but $\boxminus \rho$ does not, we conclude:

$$(u,i) \models \varphi \ . \tag{4.8}$$

Also, $i > 0$ and

$$(u,i-1) \not\models \boxminus \rho \tag{4.9}$$

because otherwise $\boxminus \rho$ would hold at i (cf. (4.5)). Hence,

$$(u,i-1) \models \varphi \; \mathsf{S} \; (\psi \wedge \rho) \ , \tag{4.10}$$

again because the second conjunct of φ_R holds at i but $\boxminus \rho$ does not. Thus there exists a position i' with $i' < i$ such that

$$(u,i') \models \psi \ , \tag{4.11}$$

$$(u,i') \models \rho \ , \tag{4.12}$$

$$(u,j) \models \varphi \qquad \text{for every } j \text{ with } i' < j < i. \tag{4.13}$$

From (4.6) and (4.8) we conclude that instead of (4.13) we even have

$$(u,j) \models \varphi \qquad \text{for every } j \text{ with } i' < j < |u|. \tag{4.14}$$

So (4.11), (4.12), and (4.14) show that i was not chosen minimal—a contradiction. ∎

4.2. Characterization of small classes. Before we can give an algebraic characterization of the levels of the since hierarchy, we need to characterize classes of languages expressible by very simple formulas.

We recall some terminology and notation, following [1]. $\mathrm{B}(1,2)^1$ stands for the semigroup $\{1, a, b\}$ whose multiplication is given by $xa = a$, $xb = b$, and $x1 = x$, for all $x \in \{1, a, b\}$. The subsemigroups $\{a, b\}$ and $\{1, a\}$ of $\mathrm{B}(1,2)^1$ are denoted by $\mathrm{B}(1,2)$ and Sl_2, respectively.

Given a class $\mathcal{C}$ of finite semigroups, $V(\mathcal{C})$ denotes the smallest pseudovariety of semigroups containing $\mathcal{C}$, and $V^*(\mathcal{C})$ the smallest pseudovariety of semigroups closed under semidirect products that contains $\mathcal{C}$. Given a pseudovariety $\mathbf{V}$ of semigroups, $\mathbf{V}^n$ stands for the iterated semidirect product $\mathbf{V} * \cdots * \mathbf{V}$ with n factors and $\mathbf{LV}$ for the class of all finite semigroups whose submonoids belong to $\mathbf{V}$.

For $n \geq 1$, $\mathbf{D}_n$ is the pseudovariety of semigroups defined by the equation

$$y x_1 \dots x_n = x_1 \dots x_n \ , \tag{4.15}$$

and

$$\mathbf{D} = \bigcup_{n>0} \mathbf{D}_n \ . \tag{4.16}$$

$\mathbf{R}$ is the pseudovariety of all $\mathcal{R}$-trivial finite semigroups, $\mathbf{SL}$ is the pseudovariety of all finite semilattices, and $\mathbf{MD}_1$ is the pseudovariety of semigroups generated by all semigroups of the form S^1 with $S \in \mathbf{D_1}$.

Equivalent descriptions of $\mathbf{D_1}$, $\mathbf{SL}$, $\mathbf{R}$, $\mathbf{MD}_1$, and $\mathbf{LR}$ are (see, e. g., [1]):

$$\mathbf{D_1} = V(\{\mathrm{B}(1{,}2)\}) \ , \tag{4.17}$$

$$\mathbf{SL} = V(\{\mathrm{Sl}_2\}) \ , \tag{4.18}$$

$$\mathbf{D} = V^*(\mathbf{D_1}) \ , \tag{4.19}$$

$$\mathbf{R} = V^*(\mathbf{SL}) \ , \tag{4.20}$$

$$\mathbf{MD}_1 = V(\{\mathrm{B}(1{,}2)^1\}) \ , \tag{4.21}$$

$$\mathbf{LR} = V^*(\{\mathrm{B}(1{,}2), \mathrm{Sl}_2\}) \ = \ \mathbf{R} * \mathbf{D} \ . \tag{4.22}$$

Equation (4.22) is from [16].

We first characterize the class of temporal logic formulas that corresponds to the pseudovariety $\mathbf{D}$.

LEMMA 4.4. $L(\mathtt{Pv}) = L(\mathbf{D})$.

Proof. A language $L \subseteq \Sigma_P^+$ belongs to $L(\mathbf{D})$ if and only if L is a boolean combination of singleton sets $\{u\}$ and languages of the form $\Sigma_P^* u$ for $u \in \Sigma_P^+$. This follows almost immediately from (4.15) and (4.16), see also [14].

For each $a \in \Sigma_P$ let α_a be a propositional formula such that $L(\alpha_P) = \Sigma_P^* a$. Assume u is a non-empty word over Σ_P^+ of length n. Then

$$L(\alpha_{u(n-1)} \wedge \ominus(\alpha_{u(n-2)} \wedge \ominus(\alpha_{u(n-3)} \wedge \ldots (\alpha_{u(0)} \wedge \neg\ominus \textsc{True})))) = \{u\} \ . \tag{4.23}$$

Similarly,

$$L(\alpha_{u(n-1)} \wedge \ominus(\alpha_{u(n-2)} \wedge \ominus(\alpha_{u(n-3)} \wedge \ldots \alpha_{u(0)}))) = \Sigma_P^* u \ . \tag{4.24}$$

Thus, $L(\mathbf{D}) \subseteq L(\mathtt{Pv})$.

For the other direction, recall that by Lemma 4.1, every formula in $\mathtt{Pv}$ is equivalent to a formula which is a boolean combination of formulas of the form $\ominus\ominus\ldots\ominus p$ or $\ominus\ominus\ldots\ominus\textsc{True}$. Now, observe that

$$L(\ominus\ominus\ldots\ominus p) = \bigcup_u \Sigma^* u \ , \tag{4.25}$$

where u ranges over all words of length $n+1$ with $p \in u(n)$ and where n is the number of times the previously operator is applied in the formula on the left-hand side. Dropping the restriction "$p \in u(n)$" leads to an expression equivalent to $\ominus\ominus\ldots\ominus\textsc{True}$. This shows $L(\mathtt{Pv}) \subseteq L(\mathbf{D})$. □

Next, we characterize $\mathbf{MD}_1$.

LEMMA 4.5.

1. *A language over A is recognized by* $\mathrm{B}(1{,}2)^1$ *if and only if it is of the form A^*BC^* for some $B, C \subseteq A$.*
2. *A language over A is recognized by a semigroup in* $\mathbf{MD}_1$ *if and only if it is a boolean combination of languages of the form A^*BC^* for sets $B, C \subseteq A$.*
3. $L(\underline{\mathtt{Sn}_1}) = L(\mathbf{MD}_1)$.

See also Lemma 1.3 in [4].

Proof. For the proof of part 1, let $h\colon A^+ \to \mathrm{B}(1{,}2)^1$ be a homomorphism. Set $M_s = h^{-1}(s) \cap A$ for every $s \in \mathrm{B}(1{,}2)^1$. Then $h^{-1}(a) = A^* M_a M_1^*$, $h^{-1}(b) = A^* M_b M_1^*$, and $h^{-1}(1) = M_1^+$. Therefore, every language recognized by $\mathrm{B}(1{,}2)^1$ is already of the form A^*BC^* for $B, C \subseteq A$. On the other hand, for every possible choice of $B, C \subseteq A$, the set A^*BC^* is recognized by $\mathrm{B}(1{,}2)^1$, as can be seen as follows. The claim is trivial if $B = \emptyset$. If $B \neq \emptyset$, consider the homomorphism $h\colon A^+ \to \mathrm{B}(1{,}2)^1$ defined by

$$h(c) = \begin{cases} a & \text{if } c \in B, \\ 1 & \text{if } c \in C \setminus B, \\ b & \text{otherwise.} \end{cases}$$

Then $h^{-1}(a) = A^*BC^*$.

For the proof of part 2 observe that since $\mathbf{MD}_1$ is generated by $\mathrm{B}(1,2)^1$, a language is recognized by an element of $\mathbf{MD}_1$ if and only if it is a boolean combination of languages recognized by $\mathrm{B}(1,2)^1$. So the claim follows from part 1.

Part 3 is a trivial consequence of part 2 in view of the following equation, where $B = \{b_1, \ldots, b_m\}$ and $C = \{c_1, \ldots, c_n\}$ are (possibly empty) subsets of A.

$$L((c_1 \vee c_2 \vee \ldots \vee c_n)\ \mathsf{S}\ (b_1 \vee b_2 \vee \ldots \vee b_m)) = A^*BC^* \tag{4.26}$$

Recall also that every formula in $\underline{\mathtt{Sn}}_1$ is equivalent to a boolean combination of formulas as in (4.26) and atomic formulas. □

The next pseudovariety we consider is $\mathbf{R}$. Let $\mathtt{WOn}$ ("W" suggesting "weak") denote the class of formulas that are boolean combinations of formulas $\diamondminus\varphi$ where φ belongs to $\mathtt{On}$. Similarly, let $\mathtt{WOn}_1$ denote the set of all formulas from $\mathtt{WOn}$ of nesting depth at most 1 in $\diamondminus$, i.e., a formula belongs to $\mathtt{WOn}_1$ if and only if it is a boolean combination of formulas $\diamondminus\varphi$ where φ is propositional.

LEMMA 4.6.

1. *A language over A^+ is recognized by Sl_2 if and only if it is of the form A^*BA^* for some $B \subseteq A$ or the complement of such a language.*
2. $L(\mathtt{WOn}_1) = L(\mathbf{SL})$.
3. $L(\mathtt{WOn}) \subseteq L(\mathbf{R})$.
4. $L(\mathbf{R}) \subseteq L(\mathtt{WOn} \star \mathtt{Pv})$.

Proof. The proof of part 1 follows the pattern of the proof of the first part of the previous lemma. Part 2 follows from part 1, as a language is a boolean combination of languages of the form A^*BA^* if and only if it is a boolean combination of languages of the form A^*bA^* for $b \in A$.

For the proof of part 3, we first show $L(\mathtt{WOn}_1^k) \subseteq L(\mathbf{SL}^k)$ for every $k > 0$. The induction base follows from part 2. For the inductive step, assume $L(\mathtt{WOn}_1^k) \subseteq L(\mathbf{SL}^k)$ for some k. Let Φ be the set of all PTL formulas φ such that $L(\varphi) \in L(\mathbf{SL}^k)$. Then, by the inductive assumption, $\mathtt{WOn}_1^k \subseteq \Phi$. Moreover,

$$\begin{array}{rcll} L(\mathtt{WOn}_1^{k+1}) & = & L(\mathtt{WOn}_1 \star \mathtt{WOn}_1^k) & \\ & \subseteq & L(\mathtt{WOn}_1 \star \Phi) & \text{definition of } \Phi \\ & \subseteq & L(\mathbf{SL} * \mathbf{SL}^k) & \text{SSP I} \\ & = & L(\mathbf{SL}^{k+1})\ . & \end{array}$$

The claim now follows from (4.20).

For the proof of part 4, we first show $L(\mathbf{SL}^k) \subseteq L(\mathtt{WOn}_1^k \star \mathtt{Pv})$ for $k > 0$. The induction base follows from part 2. For the inductive step, assume $L(\mathbf{SL}^k) \subseteq L(\mathtt{WOn}_1^k \star \mathtt{Pv})$ for some k. Let Φ be the set of all formulas $\varphi \in \mathtt{WOn}_1^k \star \mathtt{Pv}$ such that $L(\varphi) \in L(\mathbf{SL}^k)$. Then, by the inductive assumption, $L(\mathbf{SL}^k) = L(\Phi)$. Moreover,

$$\begin{array}{rcll} L(\mathbf{SL}^{k+1}) & = & L(\mathbf{SL} * \mathbf{SL}^k) & \\ & \subseteq & L(\mathtt{WOn}_1 \star \mathtt{Pv} \star \Phi) & \text{SSP II} \\ & \subseteq & L(\mathtt{WOn}_1 \star \mathtt{Pv} \star \mathtt{WOn}_1^k \star \mathtt{Pv}) & \text{definition of } \Phi \\ & = & L(\mathtt{WOn}_1 \star \mathtt{WOn}_1^k \star \mathtt{Pv}) & \text{Lemma 4.1} \\ & = & L(\mathtt{WOn}_1^{k+1} \star \mathtt{Pv})\ . & \end{array}$$

The claim of part 4 now follows from (4.20). □

It is not true that $L(\mathtt{WOn}) = L(\mathbf{R})$, and neither is it true that $L(\mathbf{R}) = L(\mathtt{WOn} \star \mathtt{Pv})$. Just note that $L(\ominus\diamondminus p) \in L(\mathbf{R}) \setminus L(\mathtt{WOn})$ and $L(p) \in L(\mathtt{WOn} \star \mathtt{Pv}) \setminus L(\mathbf{R})$. We mention without proof that $L(\mathbf{R})$ is the set of all languages that are expressible via a formula which is a boolean combination of formulas $\diamondminus\varphi$ where φ is a PTL formula using the combined operator $\ominus\diamondminus$ as the only temporal operator.

4.3. Characterization of the hierarchies. We can now give an algebraic characterization of the since hierarchy.

THEOREM 4.7 (characterization of since hierarchy). *For every $k \geq 0$,*

$$L(\mathtt{Sn}_k) = L(\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D}) \ . \tag{4.27}$$

The case $k = 0$ is Theorem 4.2 in [4]. Theorem 1.1 in Chapter 7 of [6] states $L((\mathbf{R} * \mathbf{MD}_1)^k * \mathbf{LR}) \subseteq L(\mathtt{Sn}_k)$, which is the inclusion from right to left in (4.27).

A somewhat weaker version of the inclusion from right to left is Theorem 1.1 in Chapter 7 of [6].

Proof. By induction on k, we first prove

$$L(\underline{\mathtt{Sn}}_1^k \star \mathtt{Pv}) = L(\mathbf{MD}_1^k * \mathbf{D}) \tag{4.28}$$

for $k \geq 0$.

The induction base is Lemma 4.4. For the inductive step, assume (4.28) holds for some $k \geq 0$. Clearly, $L(\underline{\mathtt{Sn}}_1^{k+1} \star \mathtt{Pv}) = L(\underline{\mathtt{Sn}}_1 \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$. From Lemma 4.5, we know $L(\underline{\mathtt{Sn}}_1) = L(\mathbf{MD}_1)$. Lemma 4.1 tells us $L(\underline{\mathtt{Sn}}_1 \star \mathtt{Pv} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv}) \subseteq L(\underline{\mathtt{Sn}}_1^{k+1} \star \mathtt{Pv})$. Using SSP, we obtain $L(\underline{\mathtt{Sn}}_1^{k+1} \star \mathtt{Pv}) = L(\mathbf{MD}_1^{k+1} * \mathbf{D})$.

The rest of the proof consists of the following chain of inclusions, where Φ stands for the set of all formulas from $\mathtt{Sn}_0$ that are recognized by an element of $\mathbf{R}$

$L(\mathtt{Sn}_k)$	$=$	$L(\mathtt{On} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	Theorem 4.2
	$=$	$L(\mathtt{WOn} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	Definition of $\star$
	$\subseteq$	$L(\Phi \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	Lemma 4.6
	$=$	$L(\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})$	SSP and (4.28)
	$\subseteq$	$L(\Phi \star \mathtt{Pv} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	SSP II
	$\subseteq$	$L(\mathtt{WOn} \star \mathtt{Pv} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	Lemma 4.6
	$=$	$L(\mathtt{WOn} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	Lemma 4.1
	$=$	$L(\mathtt{On} \star \underline{\mathtt{Sn}}_1^k \star \mathtt{Pv})$	Definition of $\star$
	$=$	$L(\mathtt{Sn}_k)$	Theorem 4.2

This completes the proof. $\square$

Using $\mathcal{C}^\rho$ to denote the class of all semigroups for which the reverse belongs to a given class $\mathcal{C}$, we immediately get:

THEOREM 4.8 (characterization of until hierarchy). *For every $k \geq 0$,*

$$L(\mathtt{Ut}_k) = L((\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})^\rho) \ . \tag{4.29}$$

4.4. Algebraic implications. In this subsection, we deviate a little from our main route. The purpose is to explain what can be concluded from the previous theorem from a semigroup-theoretic point of view. We prove a switching rule and explain the connection to the Krohn–Rhodes Theorem.

The following switching rule is slightly weaker than the switching rule from Theorem 9 in [23], which we used when we proved the decidability of the until hierarchy for the first time and which was derived from fundamental results in [16].

THEOREM 4.9. $\mathbf{MD}_1 * \mathbf{R} \subseteq \mathbf{R} * \mathbf{MD}_1 * \mathbf{D}$.

Proof. Clearly, $\mathbf{MD}_1 * \mathbf{R} \subseteq \mathbf{MD}_1 * \mathbf{R} * \mathbf{D}$. Thus, using SSP, Lemmas 4.5 and Theorem 4.7, $L(\mathbf{MD}_1 * \mathbf{R} * \mathbf{D}) \subseteq L(\underline{\mathtt{Sn}}_1 \star \mathtt{Sn}_0)$. The latter is the same as $L(\mathtt{Sn}_1)$, which, by Theorem 4.7, is identical with $L(\mathbf{R} * \mathbf{MD}_1 * \mathbf{D})$. The claim now follows from Eilenberg's variety theorem, which, among other things, states that if $\mathbf{V}$ and $\mathbf{W}$ are pseudovarieties of semigroups such that $L(\mathbf{V}) = L(\mathbf{W})$, then $\mathbf{V} = \mathbf{W}$. $\square$

The next theorem gives an alternative description of $\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D}$.

THEOREM 4.10. *For $k \geq 0$,*

$$\mathbf{LR} * (\mathbf{MD}_1 * \mathbf{LR})^k = \mathbf{R} * \mathbf{MD}_1^k * \mathbf{D} \ . \tag{4.30}$$

Proof. The case $k = 0$ is Theorem 4.22. For $k > 0$ observe that $\mathbf{LR} = \mathbf{LR} * \mathbf{LR}$ by definition. So $\mathbf{LR} * (\mathbf{MD}_1 * \mathbf{LR})^k = (\mathbf{LR} * \mathbf{MD}_1 * \mathbf{LR})^k$ for $k > 0$. By SSP, we have $L(\mathbf{LR} * \mathbf{MD}_1 * \mathbf{LR}) = L(\mathtt{Sn}_0 \star \underline{\mathtt{Sn}_1} \star \mathtt{Sn}_0)$. But the latter is the same as $L(\mathtt{Sn}_1)$. We thus conclude: $L(\mathbf{LR} * (\mathbf{MD}_1 * \mathbf{LR})^k) = L(\mathtt{Sn}_k)$. This implies the claim in view of Theorem 4.7. □

The Krohn–Rhodes Theorem states that the pseudovariety $\mathbf{A}$, the class of all finite semigroups containing no non-trivial group, is identical with $V^*(\{\mathrm{B}(1,2)^1\})$. In view of this, it is only natural to classify a semigroup S of $\mathbf{V}$ according to how often the semidirect product operation and the use of direct products of $\mathrm{B}(1,2)^1$ alternate in a decomposition of S. We get a hierarchy of pseudovarieties, the k-th level given by

$$V^*(\{\mathrm{B}(1,2), \mathrm{Sl}_2\}) * (V(\{\mathrm{B}(1,2)^1\}) * V^*(\{\mathrm{B}(1,2), \mathrm{Sl}_2\}))^k \ . \tag{4.31}$$

(Recall that $\mathrm{B}(1,2)$ and Sl_2 are the only non-trivial factors of $\mathrm{B}(1,2)^1$.) Using the notation introduced above, we can as well write $\mathbf{LR} * (\mathbf{MD}_1 * \mathbf{LR})^k$. The previous theorem now tells us that this hierarchy corresponds to the since hierarchy of temporal logic.

4.5. Decidability. Before we can prove the decidability of the levels of the since hierarchy, we need to recall some facts from finite semigroup theory.

The first lemma gives a description of $\mathbf{MD}_1$ and is folklore.

LEMMA 4.11. *A finite semigroup belongs to $\mathbf{MD}_1$ if and only if it satisfies*

$$xyx = yx \ , \tag{4.32}$$

$$x^2 = x \ . \tag{4.33}$$

Recall that a pseudovariety $\mathbf{V}$ of semigroups is called locally finite if for every $m > 0$ there exists a finite semigroup $F_m \in \mathbf{V}$ such that every semigroup from $\mathbf{V}$ generated by at most m elements is a homomorphic image of F_m. The semigroup F_m is then unique up to isomorphism, and is denoted by $F_m\mathbf{V}$.

We say a pseudovariety $\mathbf{V}$ of semigroups is *effectively locally finite* if it is locally finite and $m \mapsto F_m\mathbf{V}$ is computable.

The following proposition can be found, for instance, in [18], as Proposition 6.2, and in [1], as Theorem 10.2.1.

PROPOSITION 4.12. *If $\mathbf{V}$ and $\mathbf{W}$ are effectively locally finite pseudovarieties of semigroups, then $\mathbf{V} * \mathbf{W}$ is effectively locally finite.*

The next proposition is Theorem 5.7 from [18].

PROPOSITION 4.13 (Straubing). *If $\mathbf{V}$ is a pseudovariety of semigroups, then $S \in \mathbf{V} * \mathbf{D}$ if and only if $S \in \mathbf{V} * \mathbf{D}_{|S|}$.*

The last fact we need is an obvious consequence of results in [22] on graph congruences.

THEOREM 4.14 (Thérien,Weiss). *If $\mathbf{V}$ and $\mathbf{W}$ are pseudovarieties such that*

- $\mathbf{LV} = \mathbf{V} * \mathbf{D}$,
- $\mathbf{V}$ *is decidable, and*
- $\mathbf{W}$ *is effectively locally finite,*

then $\mathbf{V} * \mathbf{W}$ is decidable.

The decision procedure we have in mind is the obvious modification of the procedure described under (ii) in Example 2.5 of [22]. Its correctness follows from Theorem 2.8 in [22].

We can now approach the decidability of the since hierarchy.

PROPOSITION 4.15. *Let $k \geq 0$, $l \geq 1$. The pseudovariety $\mathbf{MD}_1^k * \mathbf{D}_l$ is effectively locally finite.*

Proof. From (4.15) it is clear that $F_m\mathbf{D}_l$ is finite for $m, l \geq 1$. From Lemma 4.11 it is clear that $\mathbf{MD}_1$ is effectively locally finite. So, by repeated application of Proposition 4.12, we obtain that $\mathbf{MD}_1^k * \mathbf{D}_l$ is effectively locally finite. ☐

THEOREM 4.16. *Let $k \geq 0$. The problem of determining whether a given finite semigroup S belongs to $\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D}$ is decidable.*

Proof. In order to check whether $S \in \mathbf{R} * \mathbf{MD}_1^k * \mathbf{D}$, according to Proposition 4.13, we only have to check for $S \in \mathbf{R} * \mathbf{MD}_1^k * \mathbf{D}_{|S|}$. This is effective by Theorem 4.14 and Proposition 4.15 in view of (4.22). ☐

This finally yields:

THEOREM 4.17 (decidability of until hierarchy).

1. *Let $k \geq 0$. The problems of determining whether a regular language belongs to $L(\mathtt{Ut}_k)$ is decidable.*
2. *The problem of determining the smallest number k such that a given regular language belongs to $L(\mathtt{Ut}_k)$ can be solved effectively.*

Proof. In order to determine whether $L \in L(\mathtt{Ut}_k)$ for a given regular language L, say represented by a finite automaton, one first computes its minimal automaton $\mathfrak{A}$. The next step is to compute the transformation semigroup S of $\mathfrak{A}$. Using Theorem 4.16, it is then checked whether $S \in (\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})^\rho$, which is also the answer to the question, as can be seen as follows. First, the semigroup S is isomorphic to the syntactic semigroup of L. Second, since $(\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})^\rho$ is a pseudovariety of semigroups, $L \in L((\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})^\rho)$ if and only if $S \in (\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})^\rho$. Third, $L \in L(\mathtt{Ut}_k)$ if and only if $L \in L((\mathbf{R} * \mathbf{MD}_1^k * \mathbf{D})^\rho)$ by Theorem 4.8.

The second part follows immediately from part 1, as the decision procedures described in the proof of Theorem 4.16 are uniform in k, and one can a priori check whether a given language L is at all expressible in FTL. See [9, 15, 11]. ☐

COROLLARY 4.18. *The problem of determining the smallest number k such that the language defined by a given FTL formula belongs to $L(\mathtt{Ut}_k)$ can be solved effectively.*

Proof. It suffices to note that given an FTL formula φ it is easy to compute a finite automaton that recognizes the language defined by φ, see, for instance, [25] for the case of ω-words. ☐

Obviously, analogues of the previous two results hold for the since hierarchy.

5. Extension to ω-Words.

In this chapter we extend the algebraic characterization of the until hierarchy and its decidability to the situation where temporal logic formulas are interpreted in ω-words instead of strings (which is often enough the case, especially when temporal logic is used to specify the behavior of reactive systems).

It should be clear how the definitions of Section § 2 carry over to ω-words. To distinguish between the two settings we will write $L_\omega(\varphi)$ to denote the set of all ω-models of an FTL formula and keep writing $L(\varphi)$ for the set of all string models of an FTL formula.

5.1. Regular ω-languages.

We recall some basic facts about the algebraic treatment of regular ω-languages.

Let $L \subseteq A^\omega$. The *syntactic congruence* of L, as defined in [2] and denoted by $\equiv_L$, is the binary relation on A^+ that relates u and v if and only if

$$(\forall x, y \in A^*)\,(x(uy)^\omega \in L \leftrightarrow x(vy)^\omega \in L) \tag{5.1}$$

and

$$(\forall x, y \in A^*)\,(\forall z \in A^+)\,(xuyz^\omega \in L \leftrightarrow xvyz^\omega \in L) \tag{5.2}$$

Given an arbitrary congruence $\equiv$ on A^+, we define a similarity relation $\sim$ on A^ω as follows: $\alpha \sim \beta$ iff there exist decompositions $\alpha = u_0 u_1 u_2 \ldots$ and $\beta = v_0 v_1 v_2 \ldots$ such that $u_i \equiv v_i$ for $i \geq 0$.

For a given language $L \subseteq A^\omega$, we denote by $\sim_L$ the similarity relation induced by $\equiv_L$. The following crucial property of the syntactic congruence of a regular ω-language goes back to [2] and [12].

LEMMA 5.1 (Arnold, Pécuchet). *Let $L \subseteq A^\omega$ be regular and $\sim_L$ the similarity relation induced by $\equiv_L$.*

If $\alpha \sim_L \beta$, then $\alpha \in L$ iff $\beta \in L$.

5.2. The Ehrenfeucht–Fraïssé game for FTL. As our main tool we will use a combinatorial description of the Ehrenfeucht–Fraïssé game of [8]. In fact, we will use the following definition of game equivalence.

Let $x, y \in \Sigma_P^+ \cup \Sigma_P^\omega$, $i < |x|$, and $j < |y|$.

We define $(x,i) \equiv_k^l (y,j)$ to hold if and only if the following conditions and their symmetric counterparts (where the roles of (x,i) and (y,j) are interchanged) are met.

Initial condition: $x(i) = y(j)$.

Next condition, required if $l > 0$: If $i+1 < |x|$, then $j+1 < |y|$ and $(x, i+1) \equiv_k^{l-1} (y, j+1)$.

Eventually condition, required if $l > 0$: For every i' with $i \leq i' < |x|$, there exists j' with $j \leq j' < |y|$ such that $(x,i') \equiv_k^{l-1} (y,j')$.

Until condition, required if $k > 0$: For every i' with $i \leq i' < |x|$, there exists j' with $j \leq j' < |y|$ such that:

- **(UC1)** if $i' = i$, then $j' = j$;
- **(UC2)** $(x,i') \equiv_{k-1}^l (y,j')$;
- **(UC3)** for every j'' with $j \leq j'' < j'$, there exists i'' with $i \leq i'' < i$ such that $(x,i'') \equiv_{k-1}^l (y,j'')$.

We write $x \equiv_k^l y$ for $(x,0) \equiv_k^l (y,0)$.

For a game-theoretic interpretation of $\equiv_k^l$, see [8].

To describe the connection of $\equiv_k^l$ with temporal logic, we need the notion of residual depth.

The *residual depth* of a FTL formula φ, denoted by $\mathrm{rd}(\varphi)$, is defined to be the depth of the formula in terms of the FTL operators other than U:

$$\begin{aligned}
\mathrm{rd}(\textsc{True}) &= 0, \\
\mathrm{rd}(p) &= 0, \qquad \text{for } p \in P, \\
\mathrm{rd}(\neg\varphi) &= \mathrm{rd}(\varphi)\ , \\
\mathrm{rd}(\varphi_1 \vee \varphi_2) &= \max\{\mathrm{rd}(\varphi_1), \mathrm{rd}(\varphi_2)\}\ , \\
\mathrm{rd}(\bigcirc\varphi) &= \mathrm{rd}(\varphi) + 1\ , \\
\mathrm{rd}(\Diamond\varphi) &= \mathrm{rd}(\varphi) + 1\ , \\
\mathrm{rd}(\varphi_1 \,\mathsf{U}\, \varphi_2) &= \max\{\mathrm{rd}(\varphi_1), \mathrm{rd}(\varphi_2)\}\ .
\end{aligned}$$

The set $\mathtt{Ut}_{k,l}$ is defined to be the set of all FTL formulas of until depth at most k and residual depth at most l.

The following theorem is an adapted version of Theorem 2.1 from [8].

THEOREM 5.2 (Etessami, Wilke; soundness and completeness). *Let $x, y \in \Sigma_P^+ \cup \Sigma_P^\omega$ and $k, l \geq 0$.*

Then $x \equiv_k^l y$ if and only if x and y are models of the same $\mathtt{Ut}_{k,l}$-formulas over P.

The following lemma states that game equivalence is a congruence relation, in fact, it is a Ramsey congruence in the sense of [26]. The proof is straightforward but lengthy. The interested reader is referred to our technical report [23].

LEMMA 5.3 (congruence property). *Let $k, l \geq 0$.*

1. *The binary relation $\equiv_k^l$ restricted to Σ_P^+ is a congruence relation. That is, $\equiv_k^l$ restricted to Σ_P^+ is an equivalence relation and whenever $u \equiv_k^l v$ and $u' \equiv_k^l v'$ for $u, u', v, v' \in \Sigma_P^+$, then $uu' \equiv_k^l vv'$.*
2. *Let $u, v \in \Sigma_P^+$ and $\alpha, \beta \in \Sigma_P^\omega$. If $u \equiv_k^l v$ and $\alpha \equiv_k^l \beta$, then $u\alpha \equiv_k^l v\beta$.*
3. *If $u, v \in \Sigma_P^+$ such that $u \equiv_k^l v$, then $u^\omega \equiv_k^l v^\omega$.*

To conclude this subsection about game equivalence, we note:

REMARK 2. *Let $k, l \geq 0$, $x, y \in \Sigma_P^\infty$, $i < |x|$, and $j < |y|$. Assume $u, v \in \Sigma_P^*$ and $(x, i) \equiv_k^l (y, j)$.*

Then $(ux, |u| + i) \equiv_k^l (vy, |v| + j)$.

5.3. Reduction to strings. Using game equivalence we can now characterize the levels of the until hierarchy for ω-words. We start with a lemma.

LEMMA 5.4. *Let $k, l \geq 0$ and $u, v \in \Sigma_P^+$ be such that $u \equiv_k^l v$. For all $x, y \in \Sigma_P^*$ and $z \in \Sigma_P^+$,*

$$x(uy)^\omega \equiv_k^l x(vy)^\omega \quad , \tag{5.3}$$

$$xuyz^\omega \equiv_k^l xvyz^\omega \quad . \tag{5.4}$$

Proof. Relation (5.4) follows immediately from Lemma 5.3, part 2, and Remark 2.

By part 1 of the same lemma, we obtain $uy \equiv_k^l vy$. Thus, by part 3 of that lemma, $(uy)^\omega \equiv_k^l (vy)^\omega$. Another application of part 2 finally yields $x(uy)^\omega \equiv_k^l x(vy)^\omega$. □

The next proposition states that if a regular ω-language belongs to the k-th level of the until hierarchy then so do the classes of its syntactic congruence relation.

PROPOSITION 5.5. *Let $k \geq 0$ and $L \subseteq \Sigma_P^\omega$ such that $L \in L_\omega(\mathtt{Ut}_k)$. Each class of $\equiv_L$ belongs to $L(\mathtt{Ut}_k)$.*

Proof. There exists l such that L belongs to $L_\omega(\mathtt{Ut}_{k,l})$. We claim that every class of the syntactic congruence of L belongs to $L(\mathtt{Ut}_{k,l})$.

We only need to show that, for all $u, v \in \Sigma_P^+$, if $u \equiv_k^l v$ then $u \equiv_L v$. So assume $u \equiv_k^l v$ and let $x, y \in \Sigma_P^*$ and $z \in \Sigma_P^+$.

Assume $x(uy)^\omega \in L$. From Lemma 5.4, we know $x(uy)^\omega \equiv_k^l x(vy)^\omega$. Thus, since $L \in L_\omega(\mathtt{Ut}_{k,l})$, $x(vy)^\omega \in L$. By symmetry, if $x(vy)^\omega \in L$ then $x(uy)^\omega \in L$. Similarly, using (5.4) instead of (5.3), one shows $xuyz^\omega \in L$ iff $xvyz^\omega \in L$. So (5.1) and (5.2) hold, which means $u \equiv_L v$. □

The converse of the previous proposition is also true and is the crucial observation of this section. It is more difficult to prove, and we will dedicate almost the entire rest of this section to it.

LEMMA 5.6. *Let $k, l \geq 0$, $p = k + l + 1$, and $u, v \in \Sigma_P^*$ and $w \in \Sigma_P^+$ such that $uw^\omega \equiv_k^l vw^\omega$. Then $uw^p \equiv_k^l vw^p$.*

Proof. We prove by induction on $k' + l'$ that the following is true for $k' \leq k$, $l' \leq l$ and $i, j \geq 0$: if $(uw^\omega, i) \equiv_{k'}^{l'} (vw^\omega, j)$, $i \leq |u| + (k - k' + l - l')|w|$, and $j \leq |v| + (k - k' + l - l')|w|$, then $(uw^p, i) \equiv_{k'}^{l'} (vw^p, j)$.

Write x, y, α, and β for uw^p, vw^p, uw^ω, and vw^ω.

Induction base, $k' + l' = 0$. This is trivial.

Induction step, $k' + l' > 0$. We consider the various conditions.

Initial condition. Clearly, this condition is satisfied.

Next condition. Clearly, $(\alpha, i+1) \equiv_{k'}^{l'-1} (\beta, j+1)$, hence $(x, i+1) \equiv_{k'}^{l'-1} (y, j+1)$ by induction hypothesis.

Eventually condition. Let i' be such that $i \leq i' < |x|$. We distinguish two cases.

First case, $i < |u|$. Since $(\alpha, i) \equiv_{k'}^{l'} (\beta, j)$, there exists j' such that $(\alpha, i') \equiv_{k'}^{l'} (\beta, j')$. By Remark 2, we can assume $j' < |v| + (k - k' + l - l' + 1)|w|$. Thus we can apply the induction hypothesis and find $(x, i') \equiv_{k'}^{l'-1} (y, j')$.

Second case, $i' \geq |u|$. This case is simple, as we can choose $j' = |v| + i' - |u|$.

Until condition. Let i' be such that $i \leq i' < |x|$. Since $(\alpha, i) \equiv_{k'}^{l'} (\beta, j)$, there exists j' such that (UC1)–(UC3) hold for α, β, k', and l' instead of x, y, k, and l.

By Remark 2, we can find j_1 such that $j \leq j_1 < |v| + (k - k' + l - l' + 1)|w|$, $j_1 \leq j$, and $(\beta, j') \equiv_{k'-1}^{l'} (\alpha, j_1)$. We claim that (UC1)–(UC3) then hold for k', l', x, y, and j_1 instead of k, l, x, y, and j'.

(UC1) is trivially satisfied, as we know that if $i = i'$ then $j' = j$, and that $j \leq j_1 \leq j'$. By definition of j_1, we know $(\alpha, i') \equiv_{k'-1}^{l'} (\beta, j_1)$, hence, by induction hypothesis, $(x, i') \equiv_{k'-1}^{l'} (y, j_1)$. So (UC2) is satisfied.

To verify (U3), let j'' be such that $j \leq j'' < j_1$. If $j'' \geq |v|$, we simply take $i'' = |u| + j'' - |v|$. In the other case, if $j'' < |v|$, there is i'' such that $i \leq i'' < i$ and $(\alpha, i'') \equiv_{k'-1}^{l'} (\beta, j'')$. We can assume $i'' < |u| + (k - k' + l - l' + 1)|w|$. The induction hypothesis then yields $(x, i'') \equiv_{k'-1}^{l'} (y, j'')$. ☐

We denote by $\sim_k^l$ the similarity relation induced by $\equiv_k^l$.

PROPOSITION 5.7. *Let $k, l \geq 0$, and $s, t, u, v \in \Sigma_P^+$ such that $st^\omega \equiv_k^{l+2} uv^\omega$. Then there exists $\gamma \in \Sigma_P^\omega$ such that $st^\omega \sim_k^l \gamma \sim_k^l uv^\omega$.*

Proof. Write α and β for st^ω and uv^ω, respectively.

We first argue that we can assume $t^\omega \equiv_k^l v^\omega$.

Since we have $st^\omega \equiv_k^{l+2} uv^\omega$, there exists j such that $(st^\omega, |s|) \equiv_k^{l+1} (uv^\omega, j)$. Consequently, there exists i' such that $(st^\omega, |s| + i') \equiv_k^l (uv^\omega, j + |u|)$. By Remark 2, we can assume $i' < |t|$ and $j < |v|$. Write v', v'', t', t'' for $v(0, i'')$, $v(i'', |v|)$, $t(0, j')$, and $t(j', |t|)$. Then $uv^\omega = uv'(v''v')^\omega$ and $st^\omega = st'(t''t')^\omega$, where $(v''v')^\omega \equiv_k^l (t''t')^\omega$. We can thus replace u by uv', v by $v''v'$, s by st', and t by $t''t'$ in order to achieve $t^\omega \equiv_k^l v^\omega$.

Let $p = k + l + 1$. Set $\gamma = s(t^p v^p)^\omega$. We prove that this choice is correct.

We first show $\alpha \sim_k^l \gamma$. Let $u_0 = st^p$, $v_0 = st^p$, $u_i = t^{2p}$ for $i > 0$, and $v_i = v^p t^p$ for $i > 0$. Then $\alpha = u_0 u_1 u_2 \dots$ and $\gamma = v_0 v_1 v_2 \dots$ Trivially, $u_0 \equiv_k^l v_0$. Moreover,

$$v^p t^\omega \equiv_k^l v^p v^\omega = v^\omega \equiv_k^l t^\omega = t^p t^\omega \quad , \tag{5.5}$$

where the first equivalence follows from Lemma 5.3 and the second one holds by assumption. So Lemma 5.6 yields $v^p t^p \equiv_k^l t^p t^p$, hence $u_i \equiv_k^l v_i$ for $i > 0$, thus $\alpha \sim_k^l \gamma$.

We finally show $\gamma \sim_k^l \beta$. Let $u_0 = st^p v^p$, $v_0 = uv^p$, $u_i = t^p v^p$ for $i > 0$, and $v_i = v^{2p}$ for $i > 0$. Then $\gamma = u_0 u_1 u_2 \dots$ and $\beta = v_0 v_1 v_2 \dots$ As in the previous paragraph, we obtain $u_i \equiv_k^l v_i$ for $i > 0$. In order to prove $u_0 \equiv_k^l v_0$, we first observe

$$t^\omega = st^\omega \equiv_k^l uv^\omega \quad , \tag{5.6}$$

where the first equivalence follows from Lemma 5.3 and the second one holds by assumption. From Lemma 5.6, we can thus conclude $st^p v^p \equiv_k^l uv^p$, which means $u_0 \equiv_k^l v_0$. ☐

We can finally prove:

THEOREM 5.8. **(characterization of until hierarchy over infinite sequences)** *Let $k \geq 0$ and $L \subseteq \Sigma_P^\omega$ regular. The following are equivalent.*

- *$L \in L_\omega(\mathtt{Ut}_k)$.*
- *Each equivalence class of $\equiv_L$ belongs to $L(\mathtt{Ut}_k)$.*

Proof. One direction is Proposition 5.5. For the other direction, assume that each class of $\equiv_L$ belongs to $L(\mathtt{Ut}_k)$. Since L is regular, $\equiv_L$ has only finitely many classes and there thus exists $l \geq 0$ such that each class of $\equiv_L$ belongs to $L(\mathtt{Ut}_{k,l})$. We will prove that L belongs to $L(\mathtt{Ut}_{k,l+2})$.

For $\alpha, \beta \in \Sigma_P^\omega$ such that $\alpha \equiv_k^{l+2} \beta$, we have to show that $\alpha \in L$ iff $\beta \in L$.

Since all classes of $\equiv_k^{l+2}$ on Σ_P^ω are regular ω-languages, it is enough to consider only ultimately periodic words, that is, we can assume that α and β are of the form st^ω and uv^ω, respectively. (Recall that every nonempty regular ω-language contains an ultimately periodic word.)

By Proposition 5.7, there exists γ such that $\alpha \sim_k^l \gamma \sim_k^l \beta$. Since we assume that every class of $\equiv_L$ belongs to $L_\omega(\mathtt{Ut}_k)$, $\equiv_L$ is at least as coarse as $\equiv_k^l$, which implies that we also have $\alpha \sim_L \gamma \sim_L \beta$. We can now apply Lemma 5.1 and obtain $\alpha \in L$ iff $\gamma \in L$ iff $\beta \in L$. $\square$

We can finally prove the counterpart to Theorem 4.17 and Corollary 4.18.

THEOREM 5.9 (decidability of until hierarchy over infinite sequences).

1. *Let $k \geq 0$. The problem of determining whether a given regular ω-language belongs to $L^\omega(\mathtt{Ut}_k)$ is decidable.*
2. *The problem of determining the smallest number k such that a given regular ω-language belongs to $L^\omega(\mathtt{Ut}_k)$ can be solved effectively.*
3. *The problem of determining the smallest number k such that a given FTL formula is equivalent over ω-words to an FTL formula of until depth at most k can be solved effectively.*

Proof. In order to check whether a given regular ω-language L belongs to $L^\omega(\mathtt{Ut}_k)$ one first computes a representation of its syntactic congruence as a finite automaton $\mathfrak{A}$ (on strings). The next step is to compute the transformation semigroup S of $\mathfrak{A}$. Using Theorem 4.16, one then checks whether $S \in (\mathbf{R} * \mathbf{MD}_1 * \mathbf{D})^\rho$, which is also the answer to the question, as can be seen as follows. The semigroup is equivalent to $\Sigma_P^+/\equiv_L$, the syntactic semigroup of L. Since $(\mathbf{R} * \mathbf{MD}_1 * \mathbf{D})^\rho$ is a pseudovariety of semigroups, each class of $\equiv_L$ belongs to $L((\mathbf{R} * \mathbf{MD}_1 * \mathbf{D})^\rho)$ if and only if $S \in (\mathbf{R} * \mathbf{MD}_1 * \mathbf{D})^\rho$. Now remember that Theorem 4.8 says that $L((\mathbf{R} * \mathbf{MD}_1 * \mathbf{D})^\rho) = L(\mathtt{Ut}_k)$ and that Theorem 5.8 say that $L \in L^\omega(\mathtt{Ut}_k)$ if and only if every equivalence class of $\equiv_L$ belongs to $L(\mathtt{Ut}_k)$. $\square$

6. Problems. We would like to conclude this paper with a list of problems.

PROBLEM 1. *Determine the complexity of the decision problems considered in this paper. It is not even obvious that the procedure described in the proof of Theorem 4.16 has an elementary upper bound.*

PROBLEM 2. *Extend our results to the combined until/since hierarchy.*

PROBLEM 3. *Find a description of the k-th level of the until hierarchy in terms of structural properties of the corresponding minimal automata, as for instance given in [4] for until depth 0.*

REFERENCES

[1] J. ALMEIDA, *Finite Semigroups and Universal Algebra*, vol. 3 of Series in Algebra, World Scientific, Singapore, 1995.

[2] A. ARNOLD, *A syntactic congruence for rational ω-languages*, Theoret. Comput. Sci., 39 (1985), pp. 333–335.

[3] J. A. BRZOZOWSKI AND R. KNAST, *The dot-depth hierarchy of star-free languages is infinite*, J. Comput. System Sci., 16 (1978), pp. 37–55.

[4] J. COHEN, D. PERRIN, AND J.-E. PIN, *On the expressive power of temporal logic*, J. Comput. System Sci., 46 (1993), pp. 271–294.

[5] R. S. COHEN AND J. A. BRZOZOWSKI, *Dot-depth of star-free events*, J. Comput. System Sci., 5 (1971), pp. 1–16.

[6] J. COHEN-CHESNOT, *Etude algébrique de la logique temporelle*, PhD thesis, Université Paris 6, Paris, France, Apr. 1989.

[7] A. E. EMERSON, *Temporal and modal logic*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., vol. B: Formal Methods and Semantics, Elsevier Science Publishers B. V., Amsterdam, 1990, pp. 995–1072.

[8] K. Etessami and Th. Wilke, *An until hierarchy for temporal logic*, in Proceedings 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, N. J., 1996, IEEE, pp. 108–117.
[9] J. A. W. Kamp, *Tense Logic and the Theory of Linear Order*, PhD thesis, University of California, Los Angeles, Calif., 1968.
[10] R. E. Ladner, *Application of model theoretic games to discrete linear orders and finite automata*, Inform. and Computation, 33 (1977), pp. 281–303.
[11] R. McNaughton and S. Papert, *Counter-Free Automata*, MIT Press, Cambridge, Mass., 1971.
[12] J.-P. Pécuchet, *Etude syntaxique des parties reconnaissable de mots infinis*, Theoret. Comput. Sci., 58 (1988), pp. 231–248.
[13] D. Perrin, *Recent results on automata on infinite words*, in Mathematical Foundations of Computer Science, M. P. Chytil and V. Koubek, eds., vol. 176 of Lecture Notes in Computer Science, Springer, 1984, pp. 134–148.
[14] J.-E. Pin, *Varieties of Formal Languages*, Plenum Press, New York, 1986.
[15] M. P. Schützenberger, *On finite monoids having only trivial subgroups*, Inform. and Computation, 8 (1965), pp. 190–194.
[16] P. Stiffler, Jr., *Extensions of the fundamental theorem of finite semigroups*, Adv. in Math., 11 (1973), pp. 159–209.
[17] H. Straubing, *Varieties of recognizable sets whose syntactic monoids contain solvable groups*, PhD thesis, University of California, Berkely, CA, 1978.
[18] ———, *Finite semigroup varieties of the form* $\mathbf{V} * \mathbf{D}$, J. Pure Appl. Algebra, 36 (1985), pp. 53–94.
[19] W. Thomas, *Star-free regular sets of* ω*-sequences*, Inform. and Control, 42 (1979), pp. 148–156.
[20] ———, *Classifying regular events in symbolic logic*, J. Comput. System Sci., 25 (1982), pp. 360–376.
[21] ———, *Automata on infinite objects*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., vol. B: Formal Methods and Semantics, Elsevier Science Publishers B. V., Amsterdam, 1990, pp. 134–191.
[22] D. Thérien and A. Weiss, *Graph congruences and wreath products*, J. Pure Appl. Algebra, 36 (1985), pp. 205–215.
[23] D. Thérien and Th. Wilke, *Temporal logic and semidirect products: An effective characterization of the until hierarchy*, Technical report 96-28, DIMACS, Piscataway, N. J., 1996.
[24] ———, *Temporal logic and semidirect products: An effective characterization of the until hierarchy*, in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, Burlington, Vermont, 1996, IEEE, pp. 256–263.
[25] M. Y. Vardi and P. Wolper, *Reasoning about infinite computations*, Information and Computation, 115 (1994), pp. 1–37.
[26] Th. Wilke, *An algebraic theory for regular languages of finite and infinite words*, Intern. J. Algebra and Comput., 3 (1993), pp. 447–489.