Nesting Until and Since in Linear Temporal Logic
(Full Paper)

Denis Thérien
School of Computer Science
McGill University
Montréal, Canada

Thomas Wilke
Computer Science Dept.
CAU
Kiel, Germany

June 6, 2002

Abstract

We provide an effective characterization of the "until-since hierarchy" of linear temporal logic, that is, we show how to compute for a given temporal property the minimal nesting depth in "until" and "since" required to express it. This settles the most prominent classification problem for linear temporal logic. Our characterization of the individual levels of the "until-since hierarchy" is algebraic: for each n, we present a decidable class of finite semigroups and show that a temporal property is expressible with nesting depth at most n if and only if the syntactic semigroup of the formal language associated with the property belongs to the class provided. The core of our algebraic characterization is a new description of substitution in linear temporal logic in terms of block products of finite semigroups.

1 Introduction

Linear temporal logic is the most fundamental formalism used in computer-aided verification for specifying properties of hard- and software such as integrated circuits and communication protocols [Pnu77]. Its salient feature, which explains part of its success in practice, are temporal operators modeled after natural language constructs. They express basic temporal relations, for instance, “next”, “always”, “since”, and “until”. Of all the operators, “since” and “until” are the most powerful and important ones, as they are expressively complete [Kam68] and the only binary operators. Therefore, the most natural way to classify a temporal property and to describe its complexity is by determining the nesting depth in “until” and “since” that is required to express it, that is, by determining its level in the “until-since hierarchy”.[1] In a first step towards an effective characterization of this hierarchy, its future-only version (the “until hierarchy”) was characterized effectively in [TW96]. In a further step, level 0 of the “until-since hierarchy” was shown to be decidable [TW98]. In the present paper, we finally present effective characterizations of all the other levels of the “until-since hierarchy”.

Classifying temporal properties according to their syntactical complexity was first carried out by Sistla and Zuck [SZ87, SZ93] and Cohen-Chesnot [CC89], and has since been a subject of many papers. The objective has been to determine, for a given temporal property, which temporal operators are needed to express it, that is, to characterize fragments of linear temporal logic determined by which of the temporal operators are allowed to be used. Meanwhile, all fundamental fragments have been characterized effectively, by fragments of first-order logics [EVW97, TW98], classes of automata [EW96], or classes of semigroups (monoids) [CPP93].

Technically, we characterize each level of the until-since hierarchy by a corresponding pseudovariety of semigroups. That is, for each level of the hierarchy, we provide a decidable class of finite semigroups and show that a temporal property is definable on the respective level if and only if the transformation semigroup of the minimal automaton recognizing the property defined by the formula viewed as a formal language belongs to the class provided. There are two difficult parts to this characterization: (1) to devise the right classes of semigroups, (2) to show that these classes are decidable. For the second part, we rely on deep results in the theory of finite semigroups and categories [Alm96, Ste99] (Subsection 4.3). For the first part, we proceed as follows. In a first step (Section 3), we establish a connection between substitution in temporal logic and block products of semigroups. This is reminiscent of [TW96], there are, however, fundamen-

[1] Already fairly simple fairness constraints (for instance, “a request is served k times without another request being served”) provide properties that show that the “until-since hierarchy” is strict [EW96].

tal obstacles in the two-sided framework that have to be overcome. Instead of sets of strings (the usual objects of formal language theory), we now study what we call pointed languages, which are sets of pointed strings (strings with a distinguished position). We replace the wreath product by its two-sided version, known as the block product, and prove that if Φ and Ψ are classes of formulas and $\boldsymbol{V}$ and $\boldsymbol{W}$ are pseudovarieties of semigroups such that the languages expressible in Φ are the languages recognized by the elements of $\boldsymbol{V}$ and the pointed languages expressible in Ψ are the languages recognized by the elements of $\boldsymbol{W}$, then the block product of $\boldsymbol{V}$ and $\boldsymbol{W}$ recognizes the languages definable by formulas from Φ with propositions substituted by formulas from Ψ—the block product/substitution principle (Theorem 7). We then prove a normal form theorem for each level of the until-since hierarchy that describes it as an iterated substitution of simple formula classes (Subsection 4.1), characterize these simple formula classes, and finally use the block product/substitution principle to get a characterization of the levels of the hierarchy (Subsection 4.2).

2 Notation and Fundamental Definitions

As usual, $[n] = \{1, \ldots, n\}$ for every natural number n.

Strings are sequences of letters over a finite alphabet. The length of a string x is denoted $|x|$. The i-th letter of a string x is denoted x_i, that is, $x = x_1 \ldots x_{|x|}$. A pointed string over some alphabet A is a pair (x, i) where x is a nonempty string over A and $i \in [|x|]$. The set of all pointed strings over A is denoted $A^{\#}$. It will be identified with $A^* \times A \times A^*$, and so we may write $(x_1 \ldots x_{i-1}, x_i, x_{i+1} \ldots, x_{|x|})$ for (x, i). A pointed language over some alphabet A is a subset of $A^{\#}$.

2.1 Linear Temporal Logic

A *(linear) temporal logic formula (LTL formula)* over some finite set Σ of *propositional variables* is built from the boolean constants, tt and ff, and the elements of Σ using boolean connectives and temporal operators: $\oplus$ (next), $\ominus$ (previously), $\boxplus$ (always in the future), $\boxminus$ (always in the past), $\Diamond\!\!\!\!+$ (eventually in the future), $\Diamond\!\!\!\!-$ (eventually in the past), U (until), S (since), R (dual of until), PR (dual of since), where only U, S, R, and PR are binary and the others are unary. For each of the operators except for $\oplus$ and $\ominus$, we also have a non-strict variant, denoted by $\dot{\boxplus}$, $\dot{\boxminus}$, $\dot{\diamondplus}$, $\dot{\diamondminus}$, $\dot{\mathsf{U}}$, $\dot{\mathsf{S}}$, $\dot{\mathsf{R}}$, and $\mathsf{P}\dot{\mathsf{R}}$, respectively. We write LTL_Σ for the set of all LTL formulas over Σ.

Given an LTL formula φ over some Σ and a pointed string $(x,i) \in (2^\Sigma)^\#$, we write $(x,i) \models \varphi$ for the fact that φ holds in x at position $i \in [|x|]$. In particular,

- $(x,i) \models \mathsf{tt}$, $(x,i) \not\models \mathsf{ff}$,
- $(x,i) \models p$ if $p \in x_i$,
- $(x,i) \models \varphi \mathbin{\mathsf{U}} \psi$ if there exists j with $i < j \leq |x|$ such that $(x,j) \models \psi$ and $(x,i') \models \varphi$ for all i' with $i < i' < j$.

The semantics of $\oplus$ and $\diamondplus$ is derived from this:

$$\oplus\varphi = \mathsf{ff} \mathbin{\mathsf{U}} \varphi \ , \tag{1}$$
$$\diamondplus\varphi = \mathsf{tt} \mathbin{\mathsf{U}} \varphi \ . \tag{2}$$

The non-strict versions are obtained in a straightforward way:

$$\dot{\diamondplus}\varphi = \varphi \vee \diamondplus\varphi \ , \tag{3}$$
$$\varphi \dot{\mathsf{U}} \psi = \psi \vee (\varphi \wedge \varphi \mathbin{\mathsf{U}} \psi) \ . \tag{4}$$

The dual variants of the operators defined thus far are defined by:

$$\boxplus\varphi = \neg\diamondplus\neg\varphi \ , \qquad \dot{\boxplus}\varphi = \neg\dot{\diamondplus}\neg\varphi \ , \tag{5}$$
$$\varphi \mathbin{\mathsf{R}} \psi = \neg(\neg\varphi \mathbin{\mathsf{U}} \neg\psi) \ , \qquad \varphi\dot{\mathsf{R}}\psi = \neg(\neg\varphi\dot{\mathsf{U}}\neg\psi) \ . \tag{6}$$

The semantics of $\dot{\mathsf{S}}$ is obtained from the semantics of $\dot{\mathsf{U}}$ by replacing $<$ and $\leq$ by $>$ and $\geq$, respectively, as well as $|x|$ by 1. It is perfectly symmetric to U; the other past operators are obtained just as in (1)–(6).

With every formula φ over Σ, we associate a language and a pointed language over 2^Σ:

$$L(\varphi) = \{x \in (2^\Sigma)^+ \mid (x,1) \models \varphi\} \ , \tag{7}$$
$$P(\varphi) = \{(x,i) \in (2^\Sigma)^\# \mid (x,i) \models \varphi\} \ . \tag{8}$$

We say $L(\varphi)$ and $P(\varphi)$ are *defined* by φ.

When we want to define languages or pointed languages over arbitrary alphabets (rather than alphabets of the form 2^Σ) we will use the following convention. For each alphabet A we choose once and for all a set Σ of propositional variables such that $2^{|\Sigma|-1} < |A| \leq 2^{|\Sigma|}$ and an injective mapping $\iota\colon A \to 2^{\Sigma}$, which we extend to a homomorphism $A^* \to (2^{\Sigma})^*$. For each formula $\varphi \in \mathsf{LTL}_A$, we set

$$L_A(\varphi) = \{x \in A^+ \mid (\iota(x), 1) \models \varphi\} \ , \tag{9}$$

$$P_A(\varphi) = \{(x, i) \in (2^A)^\# \mid (\iota(x), i) \models \varphi\} \ . \tag{10}$$

We say that $L_A(\varphi)$ is the language *defined* by φ, and $P_A(\varphi)$ is the pointed language defined by φ.

When, in the above situation, φ is an LTL formula over Σ we will also say that φ is a formula over A and write $\varphi \in \mathsf{LTL}_A$. If no confusion can arise, the subscript A will be dropped. Also, by abuse of notation, we will write $p \in a$ to denote $p \in \iota(a)$. Further, we will use δ_a to denote $\bigwedge_{p \in a} p \wedge \bigwedge_{p \notin a} \neg p$, that is, $(x, i) \models \delta_a$ iff $x_i = a$. And if φ is a propositional formula, we will write A_φ for $\{a \in A \mid a \models \varphi\}$.

With every class of formulas Φ and every alphabet A, we associate the set $L_A\Phi$ of all languages over A defined by formulas $\varphi \in \Phi \cap \mathsf{LTL}_A$. The mapping $A \mapsto L_A\Phi$ is denoted by $L(\Phi)$. A language L over A is expressible in Φ if $L \in L_A\Phi$. Accordingly, for pointed languages $P_A\Phi$ and $P(\Phi)$ as well as expressibility in Φ are defined. When the alphabet is obvious, we will simply write $L \in L(\Phi)$ and $P \in P(\Phi)$ for $L \in L_A\Phi$ and $P \in P_A\Phi$.

We distinguish two notions of equivalence for LTL formulas. Formulas $\varphi, \psi \in \mathsf{LTL}_\Sigma$ are *equivalent,* denoted $\varphi \equiv \psi$, if $P(\varphi) = P(\psi)$. They are *initially equivalent,* denoted $\varphi \equiv_i \psi$, if $L(\varphi) = L(\psi)$.

2.2 Substitution

A function $\sigma\colon \mathsf{LTL}_\Sigma \to \mathsf{LTL}_\Gamma$ is a *substitution* if σ maps each formula $\varphi \in \mathsf{LTL}_\Sigma$ to the formula which is obtained from φ by replacing every occurrence of a propositional variable $p \in \Sigma$ by $\sigma(p)$. Note that a substitution is fully determined by the values for the propositional variables. When Ψ is a class of LTL formulas, then $\sigma\colon \mathsf{LTL}_\Sigma \to \mathsf{LTL}_\Gamma$ is a Ψ substitution if $\sigma(p) \in \Psi$ for every $p \in \Sigma$.

When Φ and Ψ are classes of formulas, then $\Phi \circ \Psi$ is the set of all formulas which are boolean combinations of formulas from Ψ and formulas $\sigma(\varphi)$ where $\varphi \in \Phi$ is a formula over some Σ and $\sigma\colon \mathsf{LTL}_\Sigma \to \mathsf{LTL}_\Gamma$ is a Ψ substitution for some Γ. Clearly, substitution is associative, so it is not necessary to use parentheses. If Φ is a class of formula, then Φ^i is defined by $\Phi^{i+1} = \Phi^i \circ \Phi$ where Φ^0 is the set of all propositional variables.

The most basic lemma about the semantics of LTL is the analogue of the substitution lemma for first-order logic, which we describe in what follows. Let $\sigma\colon \mathsf{LTL}_\Sigma \to \mathsf{LTL}_\Gamma$ be a substitution. For each $x \in (2^\Gamma)^+$ of length n, let $\sigma^{-1}(x)$ be the string $a_1 \dots a_n$ with

$$a_i = \{p \in \Sigma \mid (x, i) \models \sigma(p)\} \ . \tag{11}$$

Lemma 1 (LTL substitution lemma) *Let φ be an LTL formula over Σ and $\sigma\colon \mathsf{LTL}_\Sigma \to \mathsf{LTL}_\Gamma$ a substitution. Then, for each $i \in [|x|]$,*

$$(x, i) \models \sigma(\varphi) \qquad \textit{iff} \qquad (\sigma^{-1}(x), i) \models \varphi \ . \tag{12}$$

Proof. The proof goes by induction on the structure of φ. When $\varphi = \mathsf{tt}$ or $\varphi = \mathsf{ff}$, there is nothing to show. When $\varphi = p$, we have, by definition, $(x, i) \models \sigma(\varphi)$ iff $p \in a_i$ (with a_i as defined in (11)) iff $(\sigma^{-1}(x), i) \models p$. The inductive step is straightforward for the boolean connectives. We consider only one of the temporal operators; the arguments for the others are similar. Assume $\varphi = \psi \mathbin{\mathsf{U}} \chi$. Note that $\sigma(\varphi) = \sigma(\psi) \mathbin{\mathsf{U}} \sigma(\chi)$. So $(x, i) \models \sigma(\varphi)$ if and only if there exists j with $i < j \leq |x|$ such that $(x, j) \models \sigma(\chi)$ and $(x, i') \models \sigma(\psi)$ for every i' with $i < i' < j$. This is, by induction hypothesis, equivalent to: there exists j with $i < j \leq |x|$ such that $(\sigma^{-1}(x), j) \models \chi$ and $(\sigma^{-1}(x), i') \models \psi$ for every i' with $i < i' < j$. This, however, is equivalent to $(\sigma^{-1}(x), i) \models \psi \mathbin{\mathsf{U}} \chi$. $\square$

2.3 Until-Since Hierarchy

The *until-since nesting depth* of a temporal formula φ, denoted $d_{\mathsf{US}}(\varphi)$, is its nesting depth in until and since. Formally, this is defined by

$$d_{\mathsf{US}}(\mathsf{tt}) = d_{\mathsf{US}}(\mathsf{ff}) = d_{\mathsf{US}}(p) = 0 \ , \tag{13}$$

$$d_{\mathsf{US}}(o_u \varphi) = d_{\mathsf{US}}(\varphi) \ , \tag{14}$$

$$d_{\mathsf{US}}(\varphi \, o_l \, \psi) = \max(d_{\mathsf{US}}(\varphi), d_{\mathsf{US}}(\psi)) \ , \tag{15}$$

$$d_{\mathsf{US}}(\varphi \, o_b \, \psi) = \max(d_{\mathsf{US}}(\varphi), d_{\mathsf{US}}(\psi)) + 1 \ , \tag{16}$$

where p is an arbitrary propositional variable, φ and ψ are arbitrary formulas, o_u stands for any of the unary operators ($\neg$, $\ominus$, $\oplus$, $\boxplus$, $\boxminus$, $\diamondplus$, $\diamondminus$, $\dot{\boxplus}$, $\dot{\boxminus}$, $\dot{\diamondplus}$,

$\dot{\diamondminus}$), o_l stands for $\vee$ or $\wedge$, and o_b stands for any until or since operator (U, S, $\dot{\mathsf{U}}$, $\dot{\mathsf{S}}$, R, PR, $\dot{\mathsf{R}}$, $\dot{\mathsf{PR}}$).

For every i, let USH_i be the set of all formulas φ with $d_{\mathsf{US}}(\varphi) \leq i$, that is, USH_i is the set of all formulas of until-since depth at most i. The *until-since hierarchy* is the following chain:

$$L(\mathsf{USH}_0) \subseteq L(\mathsf{USH}_1) \subseteq L(\mathsf{USH}_2) \subseteq \dots \tag{17}$$

The class $L(\mathsf{USH}_i)$ is called the *i-th level of the until-since hierarchy.*

In [EW96], it was shown that the until-since hierarchy is strict, that is, each of the containments is proper, while in [TW98], it was shown that $L(\mathsf{USH}_0)$ is decidable.

It should be noted that even though the above definition of the hierarchy might seem to be susceptible to which operators are included in the syntax of temporal logic, this is not at all the case. So even if only $\oplus$, $\ominus$, $\diamondplus$, $\diamondminus$, S, and U were allowed, the hierarchy would be exactly the same. Also, if any of the other binary operators suggested in the literature (Kröger's "at next" [Krö87], Lamport's "at least as long as" [Lam83], or STeP's "wait for" [MAB$^+$94]) were added and taken into account in the above definition of until-since depth ("binary depth"), then the hierarchy would not change either.

2.4 Monoids, Semigroups, and Formal Languages

For a complete treatment of the notions presented here, the reader is referred to [Eil76, Alm95].

2.4.1 General Concepts

A semigroup is a set with a binary associative operation; a monoid is a semigroup that contains a two-sided identity element. A monoid M recognizes a language $L \subseteq A^*$ if there is a homomorphism $\theta\colon A^* \to M$ and a subset $F \subseteq M$ such that $L = \theta^{-1}(F)$. Given two monoids M and N, we say that M divides N if M is a homomorphic image of a submonoid of N. The natural classification for finite monoids is in terms of pseudovarieties, i. e., classes of finite monoids that are closed under division and direct product. If $\boldsymbol{V}$ is a pseudo-variety, we write $L(\boldsymbol{V})$ for the class of languages that can be recognized with a monoid in $\boldsymbol{V}$, and we say a congruence β over A is a *$\boldsymbol{V}$-congruence* if $A^*/\beta \in \boldsymbol{V}$.

Several operations combining two pseudovarieties to yield a third one have been investigated. We here need the so-called *block product*, $\square$, a 2-sided variant of the more classical wreath product. We will be using the following characterization of this operation.

Let $h\colon A^* \to M$ be a monoid homomorphism. We define a companion function $\hat{h}$ mapping pointed strings over A to elements of $M \times A \times M$ by $\hat{h}(x,a,y) = (h(x), a, h(y))$, that is, $\hat{h}(x,i) = (h(x_1 \dots x_{i-1}), x_i, h(x_{i+1} \dots x_{|x|}))$ for every $i \in [|x|]$.

Let γ be a finite-index congruence on A^*, let $N = A^*/\gamma$ be the quotient monoid of A^* with respect to γ, and let $h\colon A^* \to N$ be the natural homomorphism. Let β be a finite-index congruence on $(N \times A \times N)^*$. The relation $\beta \square \gamma$ is the equivalence relation on A^* defined by: $x\ \beta \square \gamma\ y$ iff

1. $x\ \gamma\ y$ and
2. for all $u, v \in A^*$, $h_{u,v}(x)\ \beta\ h_{u,v}(y)$ where $h_{u,v}(z) = b_1 \dots b_{|z|}$ with $b_i = \hat{h}(uzv, |u| + i)$.

It is not hard to verify that $\beta \square \gamma$ is a congruence of finite index on A^*, see [Thé91].

Lemma 2 [Thé91] *Let $\boldsymbol{V}$ and $\boldsymbol{W}$ be pseudovarieties of monoids. The following are equivalent for a monoid $M = A^*/\alpha$.*

- *M belongs to the pseudovariety $\boldsymbol{V} \square \boldsymbol{W}$.*
- *There exist a $\boldsymbol{W}$-congruence γ over A and a $\boldsymbol{V}$-congruence β over $N \times A \times N$ with $N = A^*/\gamma$ such that α is refined by $\beta \square \gamma$.*

Note that the block product of pseudovarieties is not associative, i. e., $(\boldsymbol{V} \square \boldsymbol{W}) \square \boldsymbol{U}$ is not equal in general to $\boldsymbol{V} \square (\boldsymbol{W} \square \boldsymbol{U})$. When writing an iterated block product without parentheses, we mean the first possibility, i. e., bracketing is from left to right.

Note also that the block product of two pseudovarieties is defined to yield a pseudovariety, so the corresponding class of languages is closed under boolean combinations.

It turns out that it is sometimes more appropriate to deal with semigroups rather than with monoids, and to consider languages as subsets of A^+ rather than A^*. All notions defined above have natural correspondance in this slightly different setting. In particular, it is possible to define the block product of $\boldsymbol{V}$ and $\boldsymbol{W}$ when either one is a pseudovariety of semigroups or even when both are pseudovarieties of semigroups. The result is then a pseudovariety of semigroups and a characterization in terms of congruences can be given, in a way very similar to the pure-monoid case.

2.4.2 Concrete Pseudovarieties

The following examples of pseudo-varieties of monoids are well-known and will be used later:

$$\mathbf{MNB} = \{M \mid \forall s,t,u \in M(s^2 = s \wedge stsus = stus)\} \ , \tag{18}$$
$$\mathbf{DA} = \{M \mid \forall e,s \in M(MeM = MsM \wedge e = e^2 \to s = s^2)\} \ . \tag{19}$$

The class of languages corresponding to $\mathbf{MNB}$, $L(\mathbf{MNB})$, can be characterized as follows. For a string x, let $\alpha(x) = \{x_i \mid 1 \leq i \leq |x|\}$ be the set of letters occurring in x, also known as the alphabet of x. For every $a \in \alpha(x)$, let $\overrightarrow{\alpha}_a(x) = \min\{i \in [|x|] \mid x_i = a\}$ be the first position (reading from left to right) where a occurs. Let $i_1 < i_2 < \cdots < i_r$ be the sequence of all positions $\overrightarrow{\alpha}_a(x)$ where $a \in \alpha(x)$. Then $\overrightarrow{\alpha}(x)$ is defined by $x_{i_1}x_{i_2}\ldots x_{i_r}$. Symmetrically, $\overleftarrow{\alpha}_a(x)$ and $\overleftarrow{\alpha}(x)$ are defined. This gives us a way to define an equivalence relation for every alphabet A. We write $x \leftrightarrow_A y$ if $\overrightarrow{\alpha}(x) = \overrightarrow{\alpha}(y)$ and $\overleftarrow{\alpha}(x) = \overleftarrow{\alpha}(y)$.

The following proposition is well-known.

Proposition 1 *A language $L \subseteq A^*$ belongs to $L(\mathbf{MNB})$ iff it is a union of $\leftrightarrow_A$-classes.*

Example 1 The language (denoted by the regular expression) $a^+b(a+b)^*a$ is a union of $\leftrightarrow_{\{a,b\}}$-classes while the language a^+ba^+ is not.

The characterization of $L(\mathbf{DA})$ uses the notion of unambiguity. A regular expression of the form $A_0^*a_1A_1^*a_1\ldots a_nA_n^*$ with $A_0,\ldots,A_n \subseteq A$ and $a_1,\ldots,a_n \in A$ is said to be *unambiguous* if for any two sequences $u_0,\ldots,u_n$ and $v_0,\ldots,v_n$ with $u_i, v_i \in A_i^*$ for every $i \leq n$ and $u_i \neq v_i$ for some i, the strings $u_0a_1u_1a_1\ldots a_nu_n$ and $v_0a_1v_1a_1\ldots a_nv_n$ are distinct. A language $L \subseteq A^*$ is *unambiguous* if it is a finite disjoint union of languages which can be denoted by unambiguous expressions.

Theorem 1 ***[Sch76]*** *A language $L \subseteq A^*$ belongs to $L(\mathbf{DA})$ iff it is unambiguous.*

Theorem 9 will present a characterization of $\mathbf{DA}$ in terms of temporal logic.

The following pseudovariety of semigroups has special importance in the algebraic theory of automata and it also plays a role in our context:

$$\mathbf{LI} = \{S \mid \forall s, e \in S(e^2 = e \to ese = e)\} \ . \tag{20}$$

A language-theoretic characterization is fairly simple to obtain:

Theorem 2 (see [Eil76]) *A language $L \subseteq A^+$ belongs to $L(\mathbf{LI})$ iff it is a boolean combination of languages of the form uA^* and A^*u for $u \in A^+$.*

2.4.3 Decidability Criteria

We will be interested in deciding membership in (iterated block products of) pseudovarieties. In the following, several decidability criteria, which have been established in finite semigroup theory, will be presented. Description of these criteria requires the language of finite categories, which are natural generalizations of monoids: notions such as division, pseudovarieties, block product all have natural definitions in this new setting.

We start with some notation. Let $\boldsymbol{V}$ be a pseudovariety of monoids. The *local* of $\boldsymbol{V}$, denoted $l\boldsymbol{V}$, is the pseudovariety of all categories where the "monoid around each object" belongs to $\boldsymbol{V}$. The *global* of $\boldsymbol{V}$, denoted $g\boldsymbol{V}$, is the pseudovariety of categories generated by the elements of $\boldsymbol{V}$ viewed as one-object categories.

A pseudovariety $\boldsymbol{V}$ of monoids is *effectively locally finite* if for every n one can construct a finite monoid $M_n \in \boldsymbol{V}$ such that every $M \in \boldsymbol{V}$ generated by at most n elements is a homorphic image of M_n, that is, M_n is a free object for $\boldsymbol{V}$. The same definition applies to pseudovarieties of categories.

If $\boldsymbol{W}$ is effectively locally finite, then, given a finite monoid M, one can construct a finite category C such that $M \in \boldsymbol{V} \square \boldsymbol{W}$ if and only if $C \in gV$, see [Til87]. Accordingly, given a finite category D, one can construct a finite category C such that $D \in g\boldsymbol{V} \square g\boldsymbol{W}$ if and only if $C \in gV$. This implies directly:

Theorem 3 *If $\boldsymbol{V}$ is a pseudovariety of monoids such that $g\boldsymbol{V}$ is decidable and if $\boldsymbol{W}$ is an effectively locally finite pseudovariety of monoids, then $\boldsymbol{V} \square \boldsymbol{W}$ and $g\boldsymbol{V} \square g\boldsymbol{W}$ are decidable.*

The next theorem allows us to strengthen the above result.

Theorem 4 ([Ste99]) *If $\boldsymbol{V}$ and $\boldsymbol{W}$ are pseudovarieties of monoids, then $g\boldsymbol{V} \square g\boldsymbol{W} = g(\boldsymbol{V} \square \boldsymbol{W})$.*

So, as a consequence:

Corollary 1 *If $\boldsymbol{V}$ is a pseudovariety of monoids such that $g\boldsymbol{V}$ is decidable and if $\boldsymbol{W}$ is an effectively locally finite pseudovariety of monoids, then $g(\boldsymbol{V} \square \boldsymbol{W})$ is decidable.*

The last decision criterion we will use needs some preparation. Let $\mathbf{D}$ be the variety of semigroups given by $\{S \mid \forall s, e \in S(e^2 = e \to se = e\}$.

We will use the two following theorems.

Theorem 5 (folklore) *For every non-trivial monoid pseudovariety $\boldsymbol{V}$, $\boldsymbol{V} \square \mathbf{LI} = \boldsymbol{V} * \mathbf{D}$ where $*$ denotes the semidirect product.*

Theorem 6 ***[Str85]*** *For every pseudovariety of monoids $\boldsymbol{V}$ such that gV is decidable, $\boldsymbol{V} * \mathbf{D}$ is decidable.*

As a consequence of the above two theorems, we get:

Corollary 2 *If $\boldsymbol{V}$ is a pseudovariety of monoids such that gV is decidable, then $\boldsymbol{V} \square \mathbf{LI}$ is decidable.*

3 The Block Product/Substitution Principle

In this section, we present the block product/substitution principle, which establishes a general, fundamental relationship between block products of pseudovarieties of monoids and substitution on classes of formulas. We start with defining what it means for a pointed language to be recognized by a monoid homomorphism or simply a monoid.

We say a pointed language P over A is *recognized* by a homomorphism $h\colon A^* \to M$ if there exists a set $U \subseteq M \times A \times M$ such that $P = \hat{h}^{-1}(U)$. We say that P is recognized by M if it is recognized by a homomorphism $A^* \to M$.

Similar to above, given a class $\boldsymbol{V}$ of finite monoids, we set

$$P(\boldsymbol{V}) = \{P \mid \exists M \in \boldsymbol{V}(P \text{ is recognized by } M)\} \ . \tag{21}$$

Example 2 For the trivial monoid pseudovariety $\mathbf{I}$, which consists of the 1-element monoid only, we have $P(\mathbf{I}) = P(\mathsf{Prop})$ where Prop is the set of all propositional formulas. On the other hand, $L(\mathbf{I}) \neq P(\mathsf{Prop})$.

Remark 1 *Let $\boldsymbol{V}$ be a pseudovariety of monoids and $P \subseteq A^{\#}$. Then the following are equivalent.*

- *P is recognizable by a monoid in $\boldsymbol{V}$.*
- *P is a boolean combination of pointed languages of the form $L \times \{a\} \times A^*$ and $A^* \times \{a\} \times L$ where $a \in A$ and $L \in L(\boldsymbol{V})$.*
- *P is a finite union of languages of the form $L \times \{a\} \times L'$ where $a \in A$ and $L, L' \in L(\boldsymbol{V})$.*

In a similar fashion, recognition by semigroups and semigroup homomorphisms can be defined, and $P(\boldsymbol{V})$ can be defined for pseudovarieties of semigroups, too. The difference is as follows.

The companion mapping $\hat{h}$ for a semigroup homomorphism $h\colon A^+ \to S$ maps pointed strings over A to $S^1 \times A \times S^1$, where S^1 is the same as S when S is a monoid and else it is S augmented by a neutral element.

The main result on block products and substitution is the following.

Theorem 7 (block product/substitution principle) *Let Φ and Ψ be classes of LTL formulas and $\boldsymbol{V}$ and $\boldsymbol{W}$ pseudovarieties of monoids or semigroups such that $L(\Phi) = L(\boldsymbol{V})$, $L(\Psi) \subseteq L(\boldsymbol{V} \,\Box\, \boldsymbol{W})$, and $P(\Psi) = P(\boldsymbol{W})$, then*

$$L(\Phi \circ \Psi) = L(\boldsymbol{V} \,\Box\, \boldsymbol{W}) \ . \tag{22}$$

Proof of Theorem 7. We consider only the case where $\boldsymbol{W}$ is a pseudovariety of monoids; the proof for a pseudovariety of semigroups is analogous. We first show the containment $L(\Phi \circ \Psi) \subseteq L(\boldsymbol{V} \,\Box\, \boldsymbol{W})$. Since $L(\boldsymbol{V} \,\Box\, \boldsymbol{W})$ is closed under boolean combinations, it is enough to show that (a) $L(\psi) \in L(\boldsymbol{V} \,\Box\, \boldsymbol{W})$ holds for $\psi \in \Psi$ and that (b) $L(\sigma(\varphi)) \in L(\boldsymbol{V} \,\Box\, \boldsymbol{W})$ holds for every formula $\varphi \in L(\Phi)$ and every Ψ substitution σ.

By assumption, (a) holds. To prove (b) let φ be a formula over Σ and $\sigma\colon \mathsf{LTL}_\Sigma \to \mathsf{LTL}_\Gamma$ a Ψ substitution. By assumption, there exists a $\boldsymbol{V}$-congruence β over 2^Σ such that $L(\varphi)$ is a union of β-classes. Further, there exists a $\boldsymbol{W}$-congruence γ over 2^Γ such that the natural homomorphism

$h\colon (2^\Gamma)^* \to (2^\Gamma)^*/\gamma$ recognizes any pointed language $P(\sigma(p))$ for $p \in \Sigma$. Let $N = (2^\Gamma)^*/\gamma$ and $g\colon (N \times 2^\Gamma \times N)^* \to (2^\Sigma)^*$ the homomorphism induced by $([u]_\gamma, a, [v]_\gamma) \mapsto \{p \mid (uav, |u|+1) \models \sigma(p)\}$; this is well-defined since h recognizes every pointed language of the form $P(\sigma(p))$. Next, let $\hat\beta$ be the congruence on $(N \times 2^\Gamma \times N)^*$ defined by $X\ \hat\beta\ Y$ iff $g(X)\ \beta\ g(Y)$. The quotient monoid $(N \times A \times N)^*/\ \hat\beta$ is clearly in $\boldsymbol{V}$, and in view of Lemma 2 for the rest it is enough to show that $L(\sigma(\varphi))$ is a union of $\hat\beta\,\Box\gamma$-classes.

So assume $x\ \hat\beta\,\Box\gamma\ y$ and $x \in L(\sigma(\varphi))$. Then $h_{\epsilon,\epsilon}(x)\ \hat\beta\ h_{\epsilon,\epsilon}(y)$ and, by the substition lemma, $\sigma^{-1}(x) \models \varphi$. Note that $h_{\epsilon,\epsilon}(x) = a_1 \dots a_{|x|}$ with $a_i = ([x_1 \dots x_{i-1}]_\gamma, x_i, [x_{i+1} \dots x_{|x|}]_\gamma)$. By definition of g, this implies $g(h_{\epsilon,\epsilon}(x)) = \sigma^{-1}(x)$. Similarly, $g(h_{\epsilon,\epsilon}(y)) = \sigma^{-1}(y)$. So $h_{\epsilon,\epsilon}(x)\ \hat\beta\ h_{\epsilon,\epsilon}(y)$ implies $g(x)\ \beta\ g(y)$, thus $\sigma^{-1}(x)\ \beta\ \sigma^{-1}(y)$, and hence $\sigma^{-1}(y) \models \varphi$, which, by the substitution lemma, shows $y \in L(\sigma(\varphi))$.

For the other containment, assume L is a language over 2^Γ which is recognized by some element of $\boldsymbol{V} \,\Box\, \boldsymbol{W}$. Then, by Lemma 2, there exist congruences β and γ such that L is a union of $\beta \,\Box\, \gamma$-classes with β and γ as specified in the lemma. We only need to show that every equivalence class of $\beta \,\Box\, \gamma$ is expressible in $\Phi \circ \Psi$.

Let $h\colon A^* \to N = A^*/\gamma$ be the natural homomorphism. For every triple $\tau = (n_l, a, n_r) \in N \times A \times N$ let $\psi_\tau \in \Psi$ be a formula defining $\hat h^{-1}(\tau)$. Further, for every pair (n'_l, n'_r) and every tuple τ as above, let $Q_\tau = \{(n''_l, a, n''_r) \mid n'_l n''_l = n_l \wedge n'_r n''_r = n_r\}$ and set $\psi_{n'_l,\tau,n'_r} = \bigvee_{\tau' \in Q_\tau} \psi_{\tau'}$. Then, clearly, $h_{u,v}(x)_i = \tau$ iff $(x,i) \models \psi_{[u]_\gamma,\tau,[v]_\gamma}$, for all $u, v \in A^*$ and $i \in [|x|]$. Let $M = (N \times A \times N)^*/\beta$ be the quotient monoid and $g\colon (N \times A \times N)^* \to M$ the natural homomorphism. For every $m \in M$, let $\varphi_m \in \Phi$ be such that $L(\varphi_m) = g^{-1}(m)$. With these definitions, we can easily construct a formula defining the $\beta \,\Box\, \gamma$-class of any string x:

$$\bigvee_{n_l h(a) n_r = h(x)} \psi_{(n_l,a,n_r)} \wedge \bigwedge_{u,v} \sigma_{u,v}\big(\varphi_{g(h_{u,v}(x))}\big) \tag{23}$$

where $\sigma_{u,v}$ is the substitution determined by

$$\sigma_{u,v}(p) = \bigvee_{p \in \tau} \psi_{[u]_\gamma,\tau,[v]_\gamma}. \tag{24}$$

Note that in (23) the conjunction has only finitely many distinct conjuncts, so we can replace it by a finite conjunction. Note also that the formula on the right-hand side of (24) might not be in Ψ, for we don't know whether

Ψ is closed under boolean combinations. But since $P(\boldsymbol{W}) = P(\Psi)$ and $\boldsymbol{W}$ is a pseudovariety, we know $P(\Psi)$ is closed under boolean combinations. Thus, there exists a formula in Ψ which is equivalent to the right-hand side of (24) and we can replace it by that. The resulting formula clearly is an element of $\Phi \circ \Psi$. $\square$

Consider again the assumption of Theorem 7 which says $L(\Psi) \subseteq L(\boldsymbol{V} \,\square\, \boldsymbol{W})$. We want to derive sufficient conditions for this to hold, but before, we rephrase this condition slightly.

Given a pointed language $P \subseteq A^{\#}$, let

$$\eta(P) = \{u \mid (u, 1) \in P\} \ . \tag{25}$$

Accordingly, given a class $\mathcal{P}$ of pointed languages, let

$$\eta(\mathcal{P}) = \{L(P) \mid P \in \mathcal{P}\} \ . \tag{26}$$

With this notation,

$$L(\Phi) = \eta(P(\Phi)) \tag{27}$$

for every set Φ of temporal formulas. Thus, the assumption $L(\Psi) \subseteq L(\boldsymbol{V} \,\square\, \boldsymbol{W})$ can be rephrased as $\eta(P(\boldsymbol{W})) \subseteq L(\boldsymbol{V} \,\square\, \boldsymbol{W})$. In order to be able to state a sufficient condition for this to hold, we need some more notation. We denote the three-element monoid $\{1, a, b\}$ with two-left zeros by $(\mathrm{B}(1,2)^1)^\rho$ and its two-element subsemigroup $\{a, b\}$ by $\mathrm{B}(1,2)^\rho$. With this, the criterion reads as follows.

Lemma 3 *Let $\boldsymbol{V}$ and $\boldsymbol{W}$ be pseudovarieties of monoids or semigroups. If $\mathrm{B}(1,2)^\rho \in \boldsymbol{V}$ or $(\mathrm{B}(1,2)^1)^\rho \in \boldsymbol{V}$, then $\eta(P(\boldsymbol{W})) \subseteq L(\boldsymbol{V} \,\square\, \boldsymbol{W})$.*

Proof. We consider only the case where $\boldsymbol{W}$ is a pseudovariety of monoids; the case of semigroups is completely analogous. By Remark 1, every $L \in \eta(P(\boldsymbol{W}))$ is a finite union of languages of the form aL' where $a \in A$ and $L' \in L(\boldsymbol{W})$. Thus, it is enough to consider a language of that form. So let $L \subseteq A^*$ be a language recognized by a monoid $N = A^*/\gamma$ where γ is a $\boldsymbol{W}$-congruence and let $a \in A$. Assume $h\colon A^* \to N$ is the natural homomorphism and $F \subseteq N$ is such that $h^{-1}(F) = L'$. Now observe that $x \in aL'$ iff $h_{\epsilon,\epsilon}(x) \in \{(h(\epsilon), a, f)X \mid f \in F \wedge X \in (N \times A \times N)^*\}$ and that the syntactic monoid of this language divides $(\mathrm{B}(1,2)^1)^\rho$. The lemma follows from Lemma 2. $\square$

Similarly, we have:

Lemma 4 *Let $\boldsymbol{V}$ and $\boldsymbol{W}$ be pseudovarieties of monoids or semigroups and assume Φ is a class of LTL formulas such that $L(\Phi) = L(\boldsymbol{V})$. If* $\mathsf{Prop} \subseteq \Phi$*, then* $\eta(P(W)) \subseteq L(\boldsymbol{V} \,\Box\, \boldsymbol{W})$.

Proof. Observe that the language described in the second to last sentence of the proof of Lemma 3 is expressible in Prop. □

4 The Until-Since Hierarchy

Recall from Subsection 2.3 that the i-th level of the until-since hierarchy comprises all languages that are definable by a formula of nesting depth at most i in until and since. In this section, we describe how we prove that for each i it is decidable whether or not a given regular language (say given by a regular expression or a finite automaton) belongs to the i-th level. Our decision procedure is uniform in the sense that given L, the minimal level it belongs to can be computed. In addition, an equivalent formula of minimal nesting depth can be computed.

4.1 Iterated Substitution

To be able to apply the block product/substitution principle, we describe USH_n as an iterated substitution of simple formula classes. This amounts to providing a strong normal form theorem, which is interesting in itself.

As stated above, the notion of until-since depth does not change when in the definition of the syntax of linear temporal logic we only allow $\oplus$, $\ominus$, $\dot{\diamond\!\!\!+}$, $\dot{\diamond\!\!\!-}$, $\dot{\mathsf{U}}$, and $\dot{\mathsf{S}}$. This is why in the following lemmas only these operators are considered.

We need that $\oplus$ and $\ominus$ can be pushed all the way inside:

Lemma 5 (switching rules for $\oplus$ and $\ominus$) *For all $\varphi, \psi \in \mathsf{LTL}$,*

$$\oplus\mathsf{ff} \equiv \mathsf{ff} \qquad \oplus\neg\varphi \equiv \neg\oplus\varphi \wedge \oplus\mathsf{tt} \ , \tag{28}$$

$$\oplus(\varphi \wedge \psi) \equiv \oplus\varphi \wedge \oplus\Psi \ , \qquad \oplus(\varphi \vee \psi) \equiv \oplus\varphi \vee \oplus\Psi \ , \tag{29}$$

$$\oplus\ominus\varphi \equiv \varphi \wedge \oplus\mathsf{tt} \ , \tag{30}$$

$$\oplus(\dot{\Diamond}^{+}\varphi) \equiv \dot{\Diamond}^{+}\oplus\varphi \ , \qquad \oplus(\dot{\Diamond}^{-}\varphi) \equiv \Diamond^{-}\varphi \wedge \oplus\mathsf{tt} \ , \tag{31}$$

$$\oplus(\varphi\dot{\mathsf{U}}\psi) \equiv \oplus\varphi\dot{\mathsf{U}}\oplus\psi \ , \qquad \oplus(\varphi\dot{\mathsf{S}}\psi) \equiv (\oplus\varphi \wedge \varphi\dot{\mathsf{S}}\psi) \vee \oplus\psi \ . \tag{32}$$

Symmetric equivalences hold for $\ominus$.

This lemma is almost folklore; the proof is straightforward.

Observe that in the above equivalences, the until-since depth of the two sides is always the same.

We further need that $\Diamond^{-}$ and $\boxminus$ can always be "pulled out". We first recall what is known from the future-only case, see [TW96]:

Lemma 6 *For all LTL formulas φ, ψ, χ,*

$$(\varphi \wedge \psi)\dot{\mathsf{U}}\chi \equiv \varphi\dot{\mathsf{U}}\chi \wedge \psi\dot{\mathsf{U}}\chi \ , \tag{33}$$

$$\varphi\dot{\mathsf{U}}(\psi \vee \chi) \equiv \varphi\dot{\mathsf{U}}\psi \vee \varphi\dot{\mathsf{U}}\chi \ , \tag{34}$$

$$(\varphi \vee \dot{\Diamond}^{+}\psi)\dot{\mathsf{U}}\chi \equiv \varphi\dot{\mathsf{U}}\chi \vee \dot{\Diamond}^{+}(\dot{\Diamond}^{+}\psi \wedge \chi) \vee \dot{\Diamond}^{+}(\psi \wedge ((\varphi \vee \psi)\dot{\mathsf{U}}\chi)) \ , \tag{35}$$

$$(\varphi \vee \dot{\boxplus}\psi)\dot{\mathsf{U}}\chi \equiv (\varphi\dot{\mathsf{U}}\chi) \vee (\dot{\Diamond}^{+}\chi \wedge \dot{\boxplus}(\varphi \vee \dot{\boxplus}\psi)) \ , \tag{36}$$

$$\varphi\dot{\mathsf{U}}(\psi \wedge \dot{\Diamond}^{+}\chi) \equiv (\varphi\dot{\mathsf{U}}\psi) \wedge \dot{\Diamond}^{+}(\psi \wedge \dot{\Diamond}^{+}\chi) \ , \tag{37}$$

$$\varphi\dot{\mathsf{U}}(\psi \wedge \dot{\boxplus}\chi) \equiv \varphi\dot{\mathsf{U}}(\psi \wedge \chi) \wedge \dot{\boxplus}(\dot{\boxplus}\chi \vee (\varphi \wedge \varphi\dot{\mathsf{U}}(\psi \wedge \chi))) \ . \tag{38}$$

Symmetric versions hold for $\dot{\mathsf{S}}$. Dual versions hold for $\dot{\mathsf{R}}$ and $\mathsf{P}\dot{\mathsf{R}}$.

Note that (35) is an improvement over the respective equivalence from [TW96]; here, $\oplus$ is not involved.

In the mixed framework, we have:

Lemma 7 *For all LTL formulas φ, ψ, χ,*

$$(\varphi \vee \dot{\Diamond}^{-}\psi)\dot{\mathsf{U}}\chi \equiv \varphi\dot{\mathsf{U}}\chi \vee (\dot{\Diamond}^{-}\psi \wedge \dot{\Diamond}^{+}\chi) \vee (\varphi\dot{\mathsf{U}}\psi \wedge \dot{\Diamond}^{+}\chi) \ , \tag{39}$$

$$(\varphi \vee \dot{\boxminus}\psi)\dot{\mathsf{U}}\chi \equiv \varphi\dot{\mathsf{U}}\chi \vee \dot{\Diamond}^{+}(\chi \wedge \dot{\boxminus}(\chi \vee \dot{\boxminus}\psi)) \ , \tag{40}$$

$$\begin{aligned}\varphi \ \mathsf{U} \ (\psi \wedge \dot{\Diamond}^{-}\chi) \equiv {} & (\dot{\Diamond}^{-}\chi \wedge \varphi\dot{\mathsf{U}}\psi) \vee (\dot{\boxplus}\varphi \wedge \dot{\Diamond}^{+}(\chi \wedge \dot{\Diamond}^{+}\psi)) \\ & \vee (\neg\dot{\Diamond}^{-}\chi \wedge \dot{\Diamond}^{+}\neg\varphi \wedge \dot{\boxplus}(\varphi \vee \dot{\Diamond}^{-}(\psi \wedge \dot{\Diamond}^{-}\chi))) \ ,\end{aligned} \tag{41}$$

$$\varphi\dot{\mathsf{U}}(\psi \wedge \dot{\boxminus}\chi) \equiv \dot{\boxminus}\chi \wedge ((\varphi \wedge \chi)\dot{\mathsf{U}}\psi) \ . \tag{42}$$

Symmetric versions hold for $\dot{\mathsf{S}}$*. Dual versions hold for* R *and* PR*.*

Proof. Except for (41), the equivalences are straightforward to verify. To verify (41), just observe that the three disjuncts of the formula distinguish the following (overlapping) cases: χ holds true at present or in the past; φ holds true at present and always in the future; χ does neither hold in the present nor in the past and φ does not hold at present or at some point in the future. $\square$

Observe that in the above lemmas the until-since depth of the right-hand sides are less than or equal to the until-since depth of the left-hand sides.

The last lemma we need shows how the strict versions of our operators can be expressed in terms of the non-strict versions and $\oplus$ and $\ominus$:

Lemma 8 *For all LTL formulas* φ*,* ψ*, and* χ*,*

$$\diamondplus\varphi \equiv \oplus\dot{\diamondplus}\varphi \ , \qquad \boxplus\varphi \equiv \oplus\dot{\boxplus}\varphi \ , \tag{43}$$

$$\varphi \mathsf{~U~} \psi \equiv \oplus\varphi\dot{\mathsf{U}}\psi \ , \qquad \varphi \mathsf{~S~} \psi \equiv \oplus\varphi\dot{\mathsf{S}}\psi \ . \tag{44}$$

Symmetric equivalences hold for the past operators.

The previous lemmas imply a first normal form, as described below. We will write $\mathsf{TL}\dot{\diamondplus}$ for the class of all formulas where only $\dot{\diamondplus}$ and $\dot{\diamondminus}$ are allowed, $\dot{\mathsf{U}}\dot{\mathsf{S}}$ for the class of all formulas of until-since depth at most 1 where only $\dot{\mathsf{U}}$ and $\dot{\mathsf{S}}$ are allowed, and $\mathsf{TL}\circledpm$ for class of all boolean combinations of formulas of the form $\oplus^i p$, $\oplus^i \mathrm{tt}$, $\ominus^i p$, and $\ominus^i \mathrm{tt}$ where the superscript i indicates iteration (nesting).

Theorem 8 (first normal form for USH) *For every* n*,*

$$\mathsf{USH}_n \equiv \mathsf{TL}\dot{\diamondplus} \circ \dot{\mathsf{U}}\dot{\mathsf{S}}^n \circ \mathsf{TL}\circledpm \ . \tag{45}$$

Proof. We describe how to transform a given formula φ_0 into an equivalent formula in the above normal form.

First, rewrite φ_0 so that only non-strict operators occur, using Lemma 8. This does not increase the until-since depth. Denote the result by φ_1.

Second, rewrite φ_1 using the switching rules from Lemma 1 in the direction from left to right, until no more rule applies. This does not increase the until-since depth and results in a formula where any subformula starting with $\oplus$ or $\ominus$ is of the form $\oplus^i\chi$ or $\ominus^i\chi$ where χ is a propositional variable or tt. Denote the result by φ_2.

Let $\chi_1, \ldots, \chi_r$ be an enumeration of the maximum subformulas of φ_2 of the form $\oplus^i\chi$ or $\ominus^i\chi$ where χ is propositional or tt; allow $i = 0$. For every $i \in [r]$, replace the occurrence of χ_i in φ_2 by a new propositional variable q_i, and denote the resulting formula by φ_3. Let σ' be the $\mathsf{TL}\pm$ substitution determined by $\sigma'(q_i) = \chi_i$. Then $\sigma'(\varphi_3) = \varphi_2$. Clearly, $d_{\mathsf{US}}(\varphi_2) = d_{\mathsf{US}}(\varphi_3)$ and neither $\oplus$ nor $\ominus$ occurs in φ_3.

Third, rewrite φ_3 so that negation is only applied to the propositional variables, using de Morgan's law, removing double negation, replacing $\neg$tt by ff and $\neg$ff by tt and replacing each negated operator by its dual; apply these rules until no more rule applies. This does not increase the until-since depth. Denote the result by φ_4.

Fourth, apply the switching rules from Lemma 7 in the direction from left to right until no more rule can be applied. The resulting formula, denoted φ_5, will have no greater until-since depth and there will be no unary temporal operator inside the scope of a binary opertor.

Let $\psi_1, \ldots, \psi_s$ be an enumeration of the maximum subformulas of φ_5 starting with a binary operator or a propositional variable. For every $i \in [s]$, replace the occurrence of ψ_i in φ_5 by a new propositional variable p_i, and denote the resulting formula by φ_6. Let σ be the substitution determined by $\sigma(p_i) = \psi_i$. Then $\sigma(\varphi_6) = \varphi_5$. Clearly, $d_{\mathsf{US}}(\varphi_5) = d_{\mathsf{US}}(\varphi_6)$, no unary temporal operator occurs in any χ_i, and no binary operator occurs in φ_6.

To complete the proof, it is enough to show that φ_5 belongs to $\dot{\mathsf{U}}\dot{\mathsf{S}}^n$ where $n = d_{\mathsf{US}}(\varphi_5)$. By induction, we show that every linear temporal formula φ of until-since depth $\leq n$ and without unary temporal operators belongs to $\dot{\mathsf{U}}\dot{\mathsf{S}}n$.

The inductive base, where $n = 0$, is trivial. Recall that, by definition, $\dot{\mathsf{U}}\dot{\mathsf{S}}^0$ contains all propositional formulas. For the inductive step, let $n > 0$. Le $\psi_1, \ldots, \psi_r$ be an enumeration of the subformulas of the form $\varphi\dot{\mathsf{U}}\psi$, $\varphi\dot{\mathsf{S}}\psi$, $\varphi\dot{\mathsf{R}}\psi$, and $\varphi\mathsf{P}\dot{\mathsf{R}}\psi$ with φ and ψ propositional. For every $i \in [r]$, replace the occurrence of ψ_i in φ by a new propositional variable p_i, and denote the resulting formula by φ'. It is easy to show (by induction) that φ' belongs to $\dot{\mathsf{U}}\dot{\mathsf{S}}^{n-1}$. Let σ be the substitution determined by $\sigma(p_i) = \psi_i$. This is not directly a $\dot{\mathsf{U}}\dot{\mathsf{S}}$ substitution, but if every ψ_i starting with $\dot{\mathsf{R}}$ or $\mathsf{P}\dot{\mathsf{R}}$ is replaced by $\neg(\neg\varphi\dot{\mathsf{U}}\neg\psi)$ and $\neg(\neg\varphi\dot{\mathsf{S}}\neg\psi)$, respectively, it is. By induction hypothesis, the claim follows. $\square$

In the following we will need a slightly different, weaker normal form, which is presented next. This normal form is better explained with two more unary temporal operators, denoted $\cent$ and $\$$. Their semantics is given by $(x,i) \models \cent\varphi$ if $(x,1) \models \varphi$, and $(x,i) \models \$\varphi$ if $(x,|x|) \models \varphi$, that is, φ is evaluated in the first and last position, respectively. (Observe that $\cent\varphi$ is equivalent with $\dot{\diamondminus}(\boxminus \mathsf{ff} \wedge \varphi)$; a symmetric statement holds for $\$\varphi$.)

Let $\mathsf{TL}\diamondpm$ be the class of all LTL formulas where $\diamondplus$ and $\diamondminus$ are the only operators allowed. Further, let $\tilde{\mathsf{TL}}\circledpm$ be the class of all LTL formulas where $\oplus$, $\ominus$, $\cent$, and $\$$ are the only temporal operators allowed. Finally, a formula belongs to $\tilde{\mathsf{US}}$ if it is a boolean combination of formulas of the form φ, $\varphi \mathrel{\mathsf{U}} \psi$, $\varphi \mathrel{\mathsf{S}} \psi$, $\cent(\varphi \dot{\mathsf{U}} \psi)$, and $\$(\varphi \dot{\mathsf{S}} \psi)$, where φ and ψ are propositional formulas.

Proposition 2 (second normal form for USH) *For every i,*

$$\mathsf{USH}_n \equiv \mathsf{TL}\diamondpm \circ \tilde{\mathsf{US}}^n \circ \tilde{\mathsf{TL}}\circledpm \ . \tag{46}$$

Proof. This normal form theorem follows immediately from the first normal form theorem. Just observe that every formula in $\mathsf{TL}\diamondpm \circ \tilde{\mathsf{US}}^n \circ \tilde{\mathsf{TL}}\circledpm$ has until-since depth at most n, so $L(\mathsf{TL}\diamondpm \circ \tilde{\mathsf{US}}^n \circ \tilde{\mathsf{TL}}\circledpm) \subseteq L(\mathsf{USH}_n)$, and also note that $L(\mathsf{TL}\dot{\diamondpm}) \subseteq L(\mathsf{TL}\diamondpm)$, $P(\dot{\mathsf{U}}\dot{\mathsf{S}}) \subseteq P(\tilde{\mathsf{US}})$, and $P(\mathsf{TL}\circledpm) \subseteq P(\tilde{\mathsf{TL}}\circledpm)$. $\square$

4.2 Algebraic Characterizations

In this section we provide algebraic classifications of the small classes of formulas we have seen above and each level of the until-since hierarchy.

For $\mathsf{TL}\diamondpm$, a characterization is known:

Theorem 9 [TW98] $L(\mathsf{TL}\diamondpm) = L(\mathbf{DA})$.

Next, we characterize $\dot{\mathsf{U}}\dot{\mathsf{S}}$.

Lemma 9 $P(\dot{\mathsf{U}}\dot{\mathsf{S}}) = P(\mathbf{MNB})$.

Proof. First, we prove that the left-hand side is contained in the right-hand side. Since $P(\mathbf{MNB})$ is closed under boolean operations, it is sufficient to show that languages definable by formulas of the form φ, $\varphi\dot{\mathsf{U}}\psi$, $\varphi\dot{\mathsf{S}}\psi$, $\cent(\varphi \mathrel{\mathsf{U}} \psi)$, and $\$(\varphi \mathrel{\mathsf{S}} \psi)$ with φ and ψ propositional formulas are unions of $\leftrightarrow_A$-classes.

From Example 2, it is clear that $P(\varphi) \in P(\mathbf{MNB})$ holds for every propositional φ. Next, assume φ and ψ are propositional. Let $B = \{a \in A \mid a \models \varphi\}$ and $C = \{a \in A \mid a \models \psi\}$. Then, clearly, $(x, i) \models \varphi \mathbin{\mathsf{U}} \psi$ iff $x_{i+1} \dots x_{|x|} \in B^* C A^*$. That is,

$$P(\varphi \mathbin{\mathsf{U}} \psi) = A^* \times A \times B^* C A^* \ . \tag{47}$$

But $B^* C A^*$ is, by Theorem 1, clearly an element of $L(\mathbf{MNB})$. So, by Remark 1, $P(\varphi \mathbin{\mathsf{U}} \psi) \in P(\mathbf{MNB})$.

For $\text{¢}(\varphi \mathbin{\mathsf{U}} \psi)$, the situation is slightly more complicated. We have $(x, i) \models \text{¢}(\varphi \mathbin{\mathsf{U}} \psi)$ iff $x \in B^* C A^*$. So,

$$\begin{aligned} P(\text{¢}(\varphi \dot{\mathsf{U}} \psi)) &= B^* C A^* \times A \times A^* && (48)\\ &\cup B^* \times C \times A^* && (49)\\ &\cup B^* \times B \times B^* C A^* \ . && (50) \end{aligned}$$

Clearly, B^* is an element of $L(\mathbf{MNB})$. Thus, by Remark 1, $P(\text{¢}(\varphi \dot{\mathsf{U}} \psi) \in P(\mathbf{MNB}))$.

For $\varphi \mathbin{\mathsf{S}} \psi$ and $\$(\varphi \dot{\mathsf{S}} \psi)$, the claim follows by symmetry.

For the converse containment, let x be an arbitrary string. We will show that $P = A^* \times A \times [x]_{\leftrightarrow_A}$ is expressible in $\tilde{\mathsf{TL}}\text{\textcircled{$\pm$}}$. By symmetry and in view of Remark 1 this is enough. Let $u = \overrightarrow{\alpha}_x(A)$ and $v = \overleftarrow{\alpha}_x(A)$. Then $|u| = |v| = |\alpha(x)|$. For every $i < |u|$ let $U_i = \{u_1, \dots, u_{i-1}\}$. Similarly, for every i with $1 < i \leq |u|$ let $V_i = \{v_{i+1}, \dots, v_{|v|}\}$. Then P is defined by

$$\bigwedge_{a \in A \setminus \alpha(x)} \neg(\mathsf{tt} \mathbin{\mathsf{U}} \delta_a) \wedge \bigwedge_{i=1}^{|u|} \bigvee_{a \in U_i} \delta_a \mathbin{\mathsf{U}} \delta_{x_i} \wedge \bigwedge_{i=1}^{|v|} \$(\bigwedge_{a \in V_i} \delta_a \mathbin{\mathsf{S}} \delta_{v_i}) \ , \tag{51}$$

where the first conjunct makes sure that to the right of the current position only letters from x occur, the second conjunct makes sure the first occurrences of the letters is correct, and the third conjunct makes sure the last occurrences are correct. $\square$

Observe that $P(\text{¢}a \mathbin{\mathsf{U}} a) \notin P(\mathbf{MNB})$.

Lemma 10 $P(\tilde{\mathsf{TL}}\text{\textcircled{$\pm$}}) = P(\mathbf{LI})$.

Proof. For the containment of the left-hand side in the right-hand side assume $\varphi \in \tilde{\mathsf{TL}}\text{\textcircled{$\pm$}}$. If $\varphi = \oplus\mathsf{tt}$, then $P(\varphi) = A^* \times A \times AA^*$, but the latter is the union of all pointed languages $A^* \times A \times aA^*$ where $a \in A$. Thus, by Remark 1 and Theorem 2, this pointed language belongs to $P(\mathbf{LI})$. If $\varphi = \oplus^i p$ for a propositional variable p and $i > 0$, then $P(\varphi)$ is the union over all pointed languages $A^* \times A \times uaA^*$ where u is an arbitrary string over A of length $i - 1$ and $a \models p$. Thus, by the same argument, $P(\varphi)$ belongs to $P(\mathbf{LI})$. If, in the above case, $i = 0$, then, clearly, $P(\varphi) \in P(\mathbf{LI})$. Next, assume $\varphi = \text{¢}\oplus^i p$ and let $B = \{a \in A \mid a \models \varphi\}$. Then

$$P(\varphi) = \bigcup_{u \in A^{i-1}, a \in C} uaA^* \times A \times A^* \tag{52}$$

$$\cup \bigcup_{u \in A^i} uA^* \cap A^i \times C \times A^* \tag{53}$$

$$\cup \bigcup_{j < i-1} A^j \times A \times \bigcup_{u \in A^{i-j-1}, a \in C} uaA^* \ . \tag{54}$$

Observe that $A^i \in L(\mathbf{LI})$ because $A^i = \bigcup_{u \in A^i} uA^* \cap (A^* \setminus \bigcup_{v \in A^{i+1}} vA^*)$. So the right-hand side of (52) belongs to $P(\mathbf{LI})$. The rest follows by symmetry.

For the converse containment, it is enough to show that $A^* \times A \times uA^*$ and $A^* \times A \times A^*u \in P(\tilde{\mathsf{TL}}\pm)$ for every $u \in A^*$. Clearly,

$$A^* \times A \times uA^* = P(\bigwedge_{i=1}^{|u|} \oplus^i \delta_{u_i}) \ , \tag{55}$$

$$A^* \times A \times A^*u = P(\oplus^{|u|}\mathsf{tt} \wedge \bigwedge_{i=1}^{|u|} \$\ominus^i \delta_{u_i}) \ . \tag{56}$$

But the right-hand sides of (55) and (56) are pointed languages in $P(\tilde{\mathsf{TL}}\pm)$. □

As a consequence of the previous lemma, Theorem 9, Proposition 2, Lemma 3, and the block product/substitution principle, we obtain:

Theorem 10 (algebraic characterization of the until-since hierarchy) *For every i,*

$$L(\mathsf{USH}_n) = L(\mathbf{DA} \,\Box\, \mathbf{MNB}^n \,\Box\, \mathbf{LI}) \ . \tag{57}$$

(Recall that $\Box$ is assumed to be right-associative. So the above expression should be read as $(((\mathbf{DA} \,\Box\, \mathbf{MNB}) \,\Box\, \mathbf{MNB}) \,\Box \cdots \Box\, \mathbf{MNB}) \,\Box\, \mathbf{LI}$.)

4.3 Decidability

In this section we will show how to decide if a given language L belongs to the i-th level of the until-since hierarchy, that is, to $L(\mathsf{USH}_i)$.

Theorem 11 (decidability of the until-since hierarchy) *For every i, the n-th level of the until hierarchy is decidable, that is, given a regular language L it can be determined whether $L \in L(\mathsf{USH}_n)$.*

Proof. By Theorem 10, this is equivalent to deciding if a semigroup belongs to the semigroup variety $\boldsymbol{V}_i \,\square\, \mathbf{LI}$ where $\boldsymbol{V}_i$ is the monoid variety given by

$$\boldsymbol{V}_0 = \mathbf{DA} \ , \tag{58}$$

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n \,\square\, \mathbf{MNB} \ . \tag{59}$$

By induction on n, we first show that $g\boldsymbol{V}_n$ is decidable. For the inductive base, we use a theorem from [Alm96], which says $l\mathbf{DA} = g\mathbf{DA}$, so $g\mathbf{DA}$ and thus $\boldsymbol{V}_0$ is decidable.

In the inductive step, we can assume $g\boldsymbol{V}_n$ is decidable. Then $g\boldsymbol{V}_{n+1}$ is decidable because of Theorem 1 and because $\mathbf{MNB}$ is effectively locally finite.

The decidability of $\boldsymbol{V}_n \,\square\, \mathbf{LI}$ finally follows from Theorem 2. $\square$

5 Conclusion

With the block product/substitution principle we have provided a very powerful tool for characterizing fragments of linear temporal logic. Using this and deep results from finite semigroup theory, we have been able to provide effective characterization of all levels of the until-since hierarchy, the most natural hierarchy for classifying temporal properties. We know how to extend Theorem 11 to ω-languages for $i = 0$ and $i = 1$, and it is possible that the technique we apply in this case can be generalized to higher levels.

References

[Alm95] Jorge Almeida. *Finite Semigroups and Universal Algebra*, volume 3 of *Series in Algebra*. World Scientific, Singapore, 1995.

[Alm96] Jorge Almeida. A syntactical proof of locality of **DA**. *Internat. J. Algebra and Comput.*, 6(2):165–177, 1996.

[CC89] Joëlle Cohen-Chesnot. *Etude algébrique de la logique temporelle*. PhD thesis, Université Paris 6, Paris, France, April 1989.

[CPP93] Joëlle Cohen, Dominique Perrin, and Jean-Eric Pin. On the expressive power of temporal logic. *J. Comput. System Sci.*, 46(3):271–294, June 1993.

[Eil76] Samuel Eilenberg. *Automata, Languages, and Machines*, volume 59-B of *Pure and Applied Mathematics*. Academic Press, New York, 1976.

[EVW97] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. In *Proceedings 12th Annual IEEE Symposium on Logic in Computer Science*, pages 228–235, Warsaw, Poland, 1997. IEEE.

[EW96] Kousha Etessami and Thomas Wilke. An until hierarchy for temporal logic. In *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, pages 108–117, New Brunswick, N. J., 1996. IEEE.

[Kam68] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, Calif., 1968.

[Krö87] Fred Kröger. *Temporal Logic of Programs*. Springer, 1987.

[Lam83] Leslie Lamport. Specifying concurrent program modules. *ACM Trans. Programming Lang. Sys.*, 5(2):190–222, 1983.

[MAB+94] Zohar Manna, Anuchit Anuchitanukul, Nikolaj Bjørner, Anca Browne, Edward Chang, Michael Colón, Luca de Alfaro, Harish Devarajan, Henny Sipma, and Tomas Uribe. STeP: the Stanford Temporal Prover. Technical Report STAN-CS-TR-94-1518, Dept. of Computer Science, Stanford University, Stanford, Calif., 1994.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57, Rhode Island, Providence, 1977. IEEE Computer Society.

[Sch76] Marcel P. Schützenberger. Sur le produit de concatenation non ambigu. *Semigroup Forum*, 13:47–75, 1976.

[Ste99] Ben Steinberg. Semidirect products of categories and applications. *Journal of Pure and Applied Algebra*, 142:153–182, 1999.

[Str85] Howard Straubing. Finite semigroup varieties of the form $\boldsymbol{V} * \mathbf{D}$. *J. Pure Appl. Algebra*, 36:53–94, 1985.

[SZ87] A. Prasad Sistla and Lenore D. Zuck. On the eventuality operator in temporal logic. In *Proceedings, Symposium on Logic in Computer Science*, pages 153–166, Ithaca, New York, 22–25June 1987. The Computer Society of the IEEE.

[SZ93] A. Prasad Sistla and Lenore D. Zuck. Reasoning in a restricted temporal logic. *Inform. and Computation*, 102(2):167–195, February 1993.

[Thé91] Denis Thérien. Two-sided wreath product of categories. *Journal of Pure and Applied Algebra*, 74:307–315, 1991.

[Til87] Brat Tilson. Categories as algebra. *J. Pure Appl. Algebra*, 48:83–198, 1987.

[TW96] Denis Thérien and Thomas Wilke. Temporal logic and semidirect products: An effective characterization of the until hierarchy. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 256–263, Burlington, Vermont, 1996. IEEE.

[TW98] Denis Thérien and Thomas Wilke. Over words, two variables are as powerful as one quantifier alternation: $\mathrm{FO}^2 = \Sigma_2 \cap \Pi_2$. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 41–47, Dallas, Texas, 24–26 May 1998.