Journal of Universal Computer Science, vol. 8, no. 8 (2002), 722-736
submitted: 2/5/02, accepted: 24/7/02, appeared: 28/7/02 © *J.UCS*

Heyting Algebras and Formal Languages

Werner Kuich
(Technische Universität Wien
kuich@tuwien.ac.at)

Norbert Sauer
(University of Calgary
nsauer@math.ucalgary.ca)

Friedrich Urbanek
(Technische Universität Wien
friedrich.urbanek@tuwien.ac.at)

Abstract: By introducing a new operation, the exponentiation of formal languages, we can define Heyting algebras of formal languages. It turns out that some well known families of languages are closed under this exponentiation, e. g., the families of regular and of context-sensitive languages.

Key Words: Lattices, automata, formal languages.

Category: F.4.3

1 Introduction

Heyting [7] proposed a formalized approach to intuitionistic logic. The structures thus obtained are distributive lattices with exponentiation, that is Heyting algebras. Birkhoff [2, 3] further developed the theory of Heyting algebras from a lattice theoretic point of view. Since then Heyting algebras, also called pseudocomplemented distributive lattices with 0, have been studied quite extensively. A good exposition on Heyting algebras can be found in the book by Balbes and Dwinger [1].

Apart from applications to topology and logic, Heyting algebras appear as skeletons of topoi. See the book by Goldblatt [5]. More recently graph morphisms, in connection with Hedetniemi's conjecture, have been studied from this point of view [9]. It turns out that also formal languages under length preserving morphisms give rise to Heyting algebras. In the present paper we will describe this connection and use it to investigate some further aspects of the category of formal languages under length preserving morphisms. We will concentrate on the language theoretic point of view transferring results from lattice theory into our notation as needed.

We define a multiplication, $\times$, of formal languages, different from concatenation and we define the exponentiation of formal languages as a new operation. (Those operations coincide with the the operations of $\times$ and exponentiation in the category of formal languages whose morphisms are the length preserving morphisms of formal languages.) We will not use the categorical definitions but provide direct constructions for those operations. The length preserving morphisms are used to define equivalence classes of formal languages. Using known results from lattice theory we will show that the set of those equivalence classes forms a Heyting algebra. In this way we obtain the Heyting algebras of equivalence classes of some wellknown families of languages like regular languages, context-sensitive languages, etc. An example shows that the equivalence classes of context-free languages do not form a Heyting algebra.

The paper consists of this and three more sections. In Section 2 we introduce the basic definitions and obtain from the general theory of Heyting algebras a calculus for equivalence classes of languages, e. g., $\mathrm{L}^{\mathrm{L}_1 \times \mathrm{L}_2} = (\mathrm{L}^{\mathrm{L}_1})^{\mathrm{L}_2}$ and $\mathrm{L}^{\mathrm{L}_1 + \mathrm{L}_2} = \mathrm{L}^{\mathrm{L}_1} \times \mathrm{L}^{\mathrm{L}_2}$ and some more of the usual computation rules.

In Section 3 we prove that some wellknown families of languages are closed under exponentiation, e. g., the families of regular and of context-sensitive languages.

In the last section we begin a study of the detailed structure of the Heyting algebra of regular languages, contextsensitive languages, etc. The basic structural elements of distributive lattices are their join and meet irreducible elements. We succeed in determining the join irreducible elements; but only partial results are obtained for meet irreducibility.

It is assumed that the reader has a basic knowledge of lattice theory (see Balbes, Dwinger [1]) and formal language and automata theory (see Harrison [6]).

2 Lattices, Morphisms and Formal Languages

Throughout this paper the symbol Σ (possibly provided with indices) denotes a finite subalphabet of some infinite alphabet Σ_∞ of symbols. All morphisms $h : \Sigma_1^* \to \Sigma_2^*$ in this paper are length preserving, i.e., $h(\Sigma_1) \subseteq \Sigma_2$.

Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$. Define $L_1 \leq L_2$ if $h(L_1) \subseteq L_2$ for some morphism $h : \Sigma_1^* \to \Sigma_2^*$ and $L_1 \sim L_2$ if $L_1 \leq L_2$ and $L_2 \leq L_1$. Then $\sim$ is an equivalence relation. If $L_1 \sim L_1'$ and $L_2 \sim L_2'$ then $L_1 \leq L_2$ iff $L_1' \leq L_2'$. It follows that $\leq$ is a partial order relation on the $\sim$-equivalence classes. We denote $\sim$-equivalence classes of languages by roman letters $\mathrm{L}, \mathrm{K}, \ldots$. If $L, K, \ldots$ are languages we denote by $\mathrm{L}, \mathrm{K}, \ldots$ the $\sim$-equivalence classes containing $L, K, \ldots$, respectively.

Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$. Define $L_1 \times L_2 = \{[a_1, b_1] \ldots [a_n, b_n] \mid a_1 \ldots a_n \in L_1,\ b_1 \ldots b_n \in L_2\} \subseteq (\Sigma_1 \times \Sigma_2)^*$ and let $L_1 + L_2$ be the disjoint union of L_1 and L_2. That is the language defined as $L_1 \cup L_2$ given that $\Sigma_1 \cap \Sigma_2 = \emptyset$. If $\Sigma_1 \cap \Sigma_2 \neq \emptyset$ then create the new alphabet $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma_2\}$ and a copy $\bar{L} \subseteq \bar{\Sigma}^*$ of L_2 and take $L_1 + L_2 = L_1 \cup \bar{L}$.

It is easy to see that if $L_1 \sim L_3$ and $L_2 \sim L_4$ then $L_1 + L_2 \sim L_3 + L_4$ and $L_1 \times L_2 \sim L_3 \times L_4$. It follows that the operations $+$ and $\times$ lift consistently to $\sim$-

equivalence classes of languages. It is clear that multiplication $\times$ and addition $+$ on $\sim$-equivalence classes are commutative and associative operations. We denote the set of $\sim$-equivalence classes of languages by $\mathcal{L}$. If $\mathfrak{F}$ is a family of languages then we denote $\mathcal{L}_{\mathfrak{F}} = \{\mathrm{L} \cap \mathfrak{F} \mid L \in \mathfrak{F}\}$.

A *lattice* $(P; \leq)$ is a partially ordered set in which for every two elements $a, b \in P$ there exists a least upper bound, denoted by $a + b$, and a greatest lower bound, denoted by $a \times b$. A lattice $(P; \leq)$ is called *distributive* if the distribution law holds:

$$a \times (b + c) = a \times b + a \times c \quad \text{for all } a, b, c \in P\,.$$

Let $\overset{\circ}{1} \in \mathcal{L}$ be the $\sim$-equivalence class containing the language $\{a\}^*$ for some $a \in \Sigma_\infty$ and $\emptyset \in \mathcal{L}$ be $\sim$-the equivalence class containing the language $\emptyset$. The following properties of $(\mathcal{L}; \leq, +, \times)$ are easy to verify:

1. Let $\mathrm{L}_1, \mathrm{L}_2 \in \mathcal{L}$ then:
   (a) $\mathrm{L}_1 + \mathrm{L}_2$ is the least of all $\mathrm{L} \in \mathcal{L}$ with $\mathrm{L}_1 \leq \mathrm{L}$ and $\mathrm{L}_2 \leq \mathrm{L}$.
   (b) $\mathrm{L}_1 \times \mathrm{L}_2$ is the greatest of all $\mathrm{L} \in \mathcal{L}$ with $\mathrm{L} \leq \mathrm{L}_1$ and $\mathrm{L} \leq \mathrm{L}_2$.
2. $\mathcal{L}$ is a lattice with $\times$ as meet and $+$ as join.
3. $\overset{\circ}{1}$ is the greatest element of the lattice $\mathcal{L}$.
4. $\emptyset$ is the least element of the lattice $\mathcal{L}$.

A family $\mathfrak{F}$ of languages is called *lattice family* if $\mathfrak{F}$ is closed under isomorphism, plus $+$ and times $\times$, and contains $\emptyset$ and Σ^* for all finite $\Sigma \subset \Sigma_\infty$.

Theorem 2.1 $(\mathcal{L}; \leq, +, \times)$ *is a lattice. If* $\mathfrak{F}$ *is a lattice family of languages then* $(\mathcal{L}_{\mathfrak{F}}; \leq, +, \times)$ *is a lattice.*

The families of regular languages, context-sensitive languages and recursive languages are lattice families.

Lemma 2.1 *The family of context-free languages is no lattice family.*

Proof. $L_1 = \{a^n b^{2n} \mid n \geq 1\}$ and $L_1 = \{a^{2n} b^n \mid n \geq 1\}$ are context-free, whereas $L_1 \times L_2 = \{[a,a]^n [b,a]^n [b,b]^n \mid n \geq 1\}$ is not context-free.

Let $\Sigma = \{h \mid h : \Sigma_1 \to \Sigma_2\}$, be the set of all functions $h : \Sigma_1 \to \Sigma_2$ considered as an alphabet. This alphabet is denoted by $\Sigma_2^{\Sigma_1}$. For $f = h_1 \ldots h_n \in \Sigma^n$ and $w = a_1 \ldots a_m \in \Sigma_1^m$ define

$$f(w) = \begin{cases} h_1(a_1) \ldots h_n(a_n) & \text{if } n = m \\ \text{undefined} & \text{if } n \neq m. \end{cases}$$

(and $\epsilon(\epsilon) = \epsilon$ if $n = 0$). For $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ define

$$L_2^{L_1} = \{f \in \Sigma^* \mid f(w) \in L_2 \text{ for all } w \in L_1 \text{ for which } f(w) \text{ is defined}\}\,.$$

Observe that $L_2^{L_1}$ depends on the sets Σ_1 and Σ_2.

Example 2.1. Let $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$.

(i) $L_1 = \emptyset$, $L_2 = \emptyset$: Then $L_2^{L_1} \subseteq (\Sigma_2^{\Sigma_1})^*$. A word $f \in (\Sigma_2^{\Sigma_1})^*$ is in $L_2^{L_1}$ iff the following implication is valid: $w \in L_1 \wedge |f| = |w| \to f(w) \in L_2$. Since for no $w \in \Sigma_1^*$, $w \in L_1$ this implication is valid. Hence $L_2^{L_1} = (\Sigma_2^{\Sigma_1})^*$.

(ii) $L_1 = \emptyset$: By the same reasoning $L_2^{\emptyset} = (\Sigma_2^{\Sigma_1})^*$.

(iii) $L_2 = \emptyset$: Define $S = \{n \mid L_1 \cap \Sigma_1^n \neq \emptyset\}$. Then $L_2^{L_1} = \bigcup_{n \in \omega - S} (\Sigma_2^{\Sigma_1})^n$.

(iv) If $L_1 \leq L_2$ and $f : \Sigma_1^* \to \Sigma_2^*$ is a morphism with $f(L_1) \subseteq L_2$ then $\{f\}^* \subseteq L_2^{L_1}$. □

We will prove that the notion of exponentiation lifts to $\sim$-equivalence classes of languages. Hence for $\sim$-equivalence classes of languages L_1 and L_2 the class $\mathrm{L}_2^{\mathrm{L}_1}$ is independent of the alphabets.

For the remainder of this section, all considered languages L, L_1, L_2, L_3, L_4 are elements of a lattice family $\mathfrak{F}$ closed under exponentiation.

Lemma 2.2 *Let $h_a : \Sigma_1 \to \Sigma$, $a \in \Sigma$, be defined by $h_a(x) = a$ for all $x \in \Sigma_1$ and consider the morphism $h : \Sigma^* \to (\Sigma^{\Sigma_1})^*$ defined by $h(a) = h_a$ for all $a \in \Sigma$. Then for $L \subseteq \Sigma^*$ and $L_1 \subseteq \Sigma_1^*$, $h(L) \subseteq L^{L_1}$, i. e., $L \leq L^{L_1}$.*

Let $\Sigma_{\mathrm{const}} = \{h_a \mid a \in \Sigma\}$ and $L_{\mathrm{const}} = L^{L_1} \cap \Sigma_{\mathrm{const}}^*$. If $L \cap \Sigma^n \neq \emptyset$ for all $n \in \omega$ then $L_{\mathrm{const}} \leq L$.

Proof. Let $w = a_1 a_2 \ldots a_n \in L$. Then $h(w) = h(a_1) \ldots h(a_n) = h_{a_1} \ldots h_{a_n}$. We have to prove that $h(w) \in L^{L_1}$.

Let $v = b_1 b_2 \ldots b_n \in L_1$. We have to prove that $h(w)(v) \in L$. We calculate:

$$h(w)(v) = h_{a_1}(b_1) h_{a_2}(b_2) \ldots h_{a_n}(b_n) = a_1 a_2 \ldots a_n = w \in L.$$

Let L contain a word of length n for every $n \in \omega$ and let $g : \Sigma_{\mathrm{const}} \to \Sigma$ be the morphism defined by $g(h_a) = a$ for all $a \in \Sigma$. Let $w = h_{a_1} h_{a_2} \ldots h_{a_n} \in L_{\mathrm{const}} \subseteq L^{L_1}$. Then $g(w) = a_1 a_2 \ldots a_n$. Hence we have to prove that if $h_{a_1} h_{a_2} \ldots h_{a_n} \in L_{\mathrm{const}}$ then $a_1 a_2 \ldots a_n \in L$. Choose now a word $b_1 \ldots b_n$ of length n in L_1. Then $a_1 a_2 \ldots a_n = h_{a_1}(b_1) h_{a_2}(b_2) \ldots h_{a_n}(b_n) \in L$.

Let $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ and $g : \Sigma_1 \to \Sigma_2$. Then we say that h is a morphism of L_1 into L_2 with $h(a) = g(a)$ for all $a \in \Sigma_1$ if $h : \Sigma_1^* \to \Sigma_2^*$ is a monoid morphism defined by $h(w) = g(a_1) \ldots g(a_n)$ for $w = a_1 \ldots a_n \in \Sigma_1^*$ and $h(L_1) \subseteq L_2$.

Lemma 2.3 *Let $L_i \subseteq \Sigma_i^*$ for $1 \leq i \leq 3$. If $L_1 \leq L_3$ then $L_2^{L_1} \geq L_2^{L_3}$.*

Proof. Let h be a morphism of L_1 into L_3. Define $h' : (\Sigma_2^{\Sigma_3})^* \to (\Sigma_2^{\Sigma_1})^*$ to be the morphism given by $h'(f) = f \circ h$ for all functions f of Σ_3 into Σ_2.

Assume now that $f = f_1 \dots f_n \in L_2^{L_3}$. Then we will show that $h'(f) \in L_2^{L_1}$. This means that we have to prove that, for $w = a_1 \dots a_n \in L_1$, $h'(f)(w) \in L_2$. We calculate

$$h'(f)(w) = h'(f_1 \dots f_n)(a_1 \dots a_n) =$$
$$h'(f_1)(a_1) \dots h'(f_n)(a_n) =$$
$$f_1(h(a_1)) \dots f_n(h(a_n)) = f(h(w))\,.$$

Since $h(w) \in L_3$ and $f \in L_2^{L_3}$ we infer that $h'(f)(w) = f(h(w)) \in L_2$.

Lemma 2.4 *Let $L_i \subseteq \Sigma_i^*$ for $1 \le i \le 3$. If $L_2 \le L_3$ then $L_2^{L_1} \le L_3^{L_1}$.*

Proof. Let h be a morphism of L_2 into L_3. Define $h' : (\Sigma_2^{\Sigma_1})^* \to (\Sigma_3^{\Sigma_1})^*$ to be the morphism given by $h'(f) = h \circ f$ for all functions f of Σ_1 into Σ_2. The proof that $h'(L_2^{L_1}) \subseteq L_3^{L_1}$ is analogous to the proof of Lemma 2.3.

Corollary 2.1 *Let $L_i \subseteq \Sigma_i^*$ for $1 \le i \le 4$. Let $L_1 \sim L_2$ and $L_3 \sim L_4$. Then $L_1^{L_3} \sim L_2^{L_4}$.*

Proof. Follows from Lemma 2.3 and Lemma 2.4.

Lemma 2.5 *Let $L \subseteq \Sigma^*$, $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$. Let h be a morphism of $L_1 \times L_2$ to L. For every $a \in \Sigma_1$ let $h_a : \Sigma_2 \to \Sigma$ be the function with $h_a(b) = h(a, b)$ and $h' : \Sigma_1^* \to (\Sigma^{\Sigma_2})^*$ be the morphism defined by $h'(a) = h_a$.*

Then h' is a morphism of L_1 into L^{L_2}.

Proof. Let $w = a_1 a_2 \dots a_n \in L_1$. Then $h'(w) = h'(a_1) \dots h'(a_n) = h_{a_1} \dots h_{a_n}$. We have to prove that $h'(w) \in L^{L_2}$.

Let $v = b_1 b_2 \dots b_n \in L_2$. Then $h'(w)(v) = h_{a_1}(b_1) h_{a_2}(b_2) \dots h_{a_n}(b_n) = h(a_1, b_1) h(a_2, b_2) \dots h(a_n, b_n) \in L$.

Lemma 2.6 *Let $L \subseteq \Sigma^*$, $L_2 \subseteq \Sigma_2^*$. Then $L_2 \times L^{L_2} \le L$.*

Proof. Let the morphism $h : (\Sigma_2 \times \Sigma^{\Sigma_2})^* \to \Sigma^*$ be given by $h(x, f) = f(x)$ for all $x \in \Sigma_2$, $f \in \Sigma^{\Sigma_2}$. It is easy to verify that $h(L_2 \times L^{L_2}) \subseteq L$. Hence $L_2 \times L^{L_2} \le L$.

Lemma 2.7 *Let $L \subseteq \Sigma^*$, $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$. Then*

$$L_1 \times L_2 \le L \text{ if and only if } L_1 \le L^{L_2}.$$

Proof. (i) Let $L_1 \times L_2 \le L$. It follows from Lemma 2.5 that then $L_1 \le L^{L_2}$.

(ii) Let $L_1 \le L^{L_2}$. Then $L_1 \times L_2 \le L^{L_2} \times L_2 \le L$ according to Lemma 2.6.

Corollary 2.2 *Let $\mathfrak{F}$ be a lattice family of languages closed under exponentiation, and L_2 and L be two elements in $\mathcal{L}_{\mathfrak{F}}$. Then $\mathrm{L}^{\mathrm{L}_2}$ is the greatest of all elements $\mathrm{L}_1 \in \mathcal{L}_{\mathfrak{F}}$ with $\mathrm{L}_1 \times \mathrm{L}_2 \le \mathrm{L}$.*

A lattice $(P, \leq)$ is called *Heyting algebra* if (i) for all $a, b \in P$ there exists a greatest $c \in P$ such that $a \times c \leq b$. This element c is denoted by b^a. (ii) There exists a least element 0 in P.

It follows from Balbes, Dwinger [1], page 173, Definition 1 and page 174, Definition 2 that $(\mathcal{L}; \leq, +, \times)$ is a Heyting algebra where the class $\emptyset$ is the 0-element. (We write L_1^L instead of $L \to L_1$ in Balbes, Dwinger [1].)

Theorem 2.2 *Let $\mathfrak{F}$ be a lattice family of languages closed under exponentiation. Then $(\mathcal{L}_{\mathfrak{F}}; \leq, +, \times)$ is a Heyting algebra where the class $\emptyset$ is the 0-element.*

Corollary 2.3 *Let $\mathfrak{F}$ be a lattice family of languages closed under exponentiation. Then, for all $\mathrm{L}_1, \mathrm{L}_2, \mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$:*

(1) If $\sum_{\mathrm{S} \in \mathcal{S}} \mathrm{S}$ exists for some subset $\mathcal{S}$ of $\mathcal{L}_{\mathfrak{F}}$ then $\sum_{\mathrm{S} \in \mathcal{S}} (\mathrm{L} \times \mathrm{S})$ exists and $\mathrm{L} \times \sum_{\mathrm{S} \in \mathcal{S}} \mathrm{S} = \sum_{\mathrm{S} \in \mathcal{S}} (\mathrm{L} \times \mathrm{S})$.

(2) $(\mathcal{L}_{\mathfrak{F}}; \leq, +, \times)$ is a distributive lattice with 0-element $\emptyset$ and 1-element $\mathring{1}$.

(3) $\mathrm{L}_1 + \mathrm{L}_2 = \mathrm{L}_2 + \mathrm{L}_1$ *and* $\mathrm{L}_1 \times \mathrm{L}_2 = \mathrm{L}_2 \times \mathrm{L}_1$.

(4) $\mathrm{L} \times (\mathrm{L}_1 + \mathrm{L}_2) = \mathrm{L} \times \mathrm{L}_1 + \mathrm{L} \times \mathrm{L}_2$ *and* $\mathrm{L} + \mathrm{L}_1 \times \mathrm{L}_2 = (\mathrm{L} + \mathrm{L}_1) \times (\mathrm{L} + \mathrm{L}_2)$.

(5) $\mathrm{L}_1 \times \mathrm{L}^{\mathrm{L}_1} \leq \mathrm{L}$.

(6) $\mathrm{L}_1 \leq \mathrm{L}^{\mathrm{L}_2}$ *if and only if* $\mathrm{L}_1 \times \mathrm{L}_2 \leq \mathrm{L}$.

(7) $\mathrm{L}_1 \leq \mathrm{L}$ *if and only if* $\mathrm{L}^{\mathrm{L}_1} = \mathring{1}$.

(8) $\mathrm{L}_1 \leq \mathrm{L}_2$ *implies* $\mathrm{L}^{\mathrm{L}_1} \geq \mathrm{L}^{\mathrm{L}_2}$ *and* $\mathrm{L}_1^{\mathrm{L}} \leq \mathrm{L}_2^{\mathrm{L}}$.

(9) $\mathrm{L}^{\mathrm{L}_1 + \mathrm{L}_2} = \mathrm{L}^{\mathrm{L}_1} \times \mathrm{L}^{\mathrm{L}_2}$ *and* $\left(\mathrm{L}^{\mathrm{L}_1}\right)^{\mathrm{L}_2} = \mathrm{L}^{\mathrm{L}_1 \times \mathrm{L}_2}$ *and* $(\mathrm{L}_1 \times \mathrm{L}_2)^{\mathrm{L}} = \mathrm{L}_1^{\mathrm{L}} \times \mathrm{L}_2^{\mathrm{L}}$.

(10) $\mathrm{L}_1 \times \mathrm{L}^{\mathrm{L}_1} = \mathrm{L}_1 \times \mathrm{L}$.

(11) $\mathrm{L}_1 \times \mathrm{L}^{\mathrm{L}_2} = \mathrm{L}_1 \times (\mathrm{L}_1 \times \mathrm{L})^{\mathrm{L}_1 \times \mathrm{L}_2}$.

(12) $\mathrm{L} \times \mathring{1} = \mathrm{L}$ *and* $\mathrm{L}^{\mathring{1}} = \mathrm{L}$ *and* $\mathring{1}^{\mathrm{L}} = \mathring{1}$.

(13) $\mathrm{L} \times \emptyset = \emptyset$ *and* $\mathrm{L} + \emptyset = \mathrm{L}$ *and* $\mathrm{L}^{\emptyset} = \mathring{1}$.

(14) $\mathrm{L}_1 \leq \mathrm{L}^{\left(\mathrm{L}^{\mathrm{L}_1}\right)}$ *and if* $\mathrm{L}_1 = \mathrm{L}^{\mathrm{L}_2}$ *then* $\mathrm{L}_1 = \mathrm{L}^{\left(\mathrm{L}^{\mathrm{L}_1}\right)}$ *for some* L_2.

Proof. Item (1) follows from Balbes, Dwinger [1], page 174, point (2). Item (2) follows from item (1) and items (3) and (4) follow from item (2) (see Balbes, Dwinger [1], page 48, section 5.)

Items (5) to (11) are Theorem 3 of Balbes, Dwinger [1] on page 174.

Items (12) and (13) are obvious.

The relation $\mathrm{L}_1 \leq \mathrm{L}^{\left(\mathrm{L}^{\mathrm{L}_1}\right)}$ of item (14) follows from items (5) and (6). If $\mathrm{L}_1 = \mathrm{L}^{\mathrm{L}_2}$ then $\mathrm{L}_2 \leq \mathrm{L}^{\left(\mathrm{L}^{\mathrm{L}_2}\right)}$ implies, according to Lemma 2.3, that $\mathrm{L}_1 = \mathrm{L}^{\mathrm{L}_2} \geq \mathrm{L}^{\left(\mathrm{L}^{\left(\mathrm{L}^{\mathrm{L}_2}\right)}\right)} = \mathrm{L}^{\left(\mathrm{L}^{\mathrm{L}_1}\right)}$.

3 Exponentiation in some important families of languages

In this section we investigate for some families of languages whether or not they are closed under exponentiation. First we show that the family of regular languages is closed under exponentiation.

Theorem 3.1 *Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ be regular languages and $\Sigma = \{h \mid h : \Sigma_1 \to \Sigma_2\}$. Then $L_2^{L_1} \subseteq \Sigma^*$ is again a regular language.*

Proof. Let $\mathfrak{A}_i = (Q_i, \Sigma_i, \delta_i, q_0^i, F_i)$ be finite automata with $||\mathfrak{A}_i|| = L_i$, $i = 1, 2$. Consider now the automaton $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ where

(i) $Q = \mathfrak{P}(\{q_2^{q_1} \mid q_1 \in Q_1,\ q_2 \in Q_2\})$,
(ii) $\delta(p, h) = \{\delta_2(q_2, h(a))^{\delta_1(q_1,a)} \mid q_2^{q_1} \in p,\ a \in \Sigma_1\}$ for $p \in Q$, $h \in \Sigma$,
(iii) $q_0 = q_0^{2\,q_0^1}$, and
(iv) $F = \{p \in Q \mid q_1 \in F_1$ implies $q_2 \in F_2$ for all $q_2^{q_1} \in p\}$.

We will show that the behavior of $\mathfrak{A}$ is $L_2^{L_1}$. For that purpose we show first that for $p \in Q$ and $f \in \Sigma^*$

$$\delta(p, f) = \{\delta_2(q_2, f(w))^{\delta_1(q_1,w)} \mid q_2^{q_1} \in p,\ w \in \Sigma^{|f|}\}\,.$$

The proof is by induction on $|f|$: (i) If $|f| = 0$, i. e., $f = \varepsilon$, we have

$$\begin{aligned}&\delta(p, \varepsilon) = p = \{q_2^{q_1} \mid q_2^{q_1} \in p\} = \\ &\{\delta_2(q_2, f(\varepsilon))^{\delta_1(q_1,\varepsilon)} \mid q_2^{q_1} \in p\} = \\ &\{\delta_2(q_2, f(w))^{\delta_1(q_1,w)} \mid q_2^{q_1} \in p,\ w \in \Sigma^0\}\,.\end{aligned}$$

(ii) If $|f| > 0$, i. e., $f = hf'$, $h \in \Sigma$, $f' \in \Sigma^*$, we have

$$\begin{aligned}\delta(p, f) &= \delta(p, hf') = \delta(\delta(p, h), f') \\ &= \{\delta_2(q_2', f'(w))^{\delta_1(q_1',w)} \mid q_2'^{q_1'} \in \delta(p, h),\ w \in \Sigma_1^{|f'|}\} \\ &= \{\delta_2(\delta_2(q_2, h(a)), f'(w))^{\delta_1(\delta_1(q_1,a),w)} \mid q_2{}^{q_1} \in p,\ a \in \Sigma_1,\ w \in \Sigma_1^{|f'|}\} \\ &= \{\delta_2(q_2, hf'(aw))^{\delta_1(q_1,aw)} \mid q_2{}^{q_1} \in p,\ a \in \Sigma_1,\ w \in \Sigma_1^{|f'|}\} \\ &= \{\delta_2(q_2, f(w))^{\delta_1(q_1,w)} \mid q_2{}^{q_1} \in p,\ w \in \Sigma_1^{|f|}\}\end{aligned}$$

(Here the third equality holds by induction hypothesis, the fourth one by definition of δ, since $q_2'^{q_1'} \in \delta(p, h)$ means that $q_2' = \delta_2(q_2, h(a))$ and $q_1' = \delta_1(q_1, a))$ for some $q_2{}^{q_1} \in p$ and $a \in \Sigma_1$.)

Now we are able to show that $||\mathfrak{A}|| = L_2^{L_1}$:

$$\begin{aligned}f \in ||\mathfrak{A}|| &\Leftrightarrow \delta(q_0, f) = \delta(\{q_0^{2\,q_0^1}\}, f) \in F \\ &\Leftrightarrow \{\delta_2(q_0^2, f(w))^{\delta_1(q_0^1,w)} \mid w \in \Sigma_1^{|f|}\} \in F \\ &\Leftrightarrow \delta_1(q_0^1, w) \in F_1 \text{ implies } \delta_2(q_0^2, f(w)) \in F_2 \text{ for all } w \in \Sigma_1^{|f|} \\ &\Leftrightarrow w \in L_1 \text{ implies } f(w) \in L_2 \text{ for all } w \in \Sigma_1^{|f|} \\ &\Leftrightarrow f \in L_2^{L_1}\end{aligned}$$

The following example shows that the family of context-free languages is *not* closed under exponentiation:

Example 3.1. Let $\Sigma_1 = \Sigma_2 = \{a, b\}$, $L_1 = \{a^n b^{2n} \mid n \geq 1\}$, and $L_2 = \{a^{2n} b^n \mid n \geq 1\}$. Let furthermore denote g_0, g_1, g_2, g_3 the four possible mappings from Σ_1 to Σ_2:

$$\begin{array}{llll} g_0(a) = a & g_1(a) = b & g_2(a) = a & g_3(a) = b \\ g_0(b) = b & g_1(b) = a & g_2(b) = a & g_3(b) = b \end{array}$$

Then $f \in L_2^{L_1} \cap \Sigma^{3n}$ iff $f(a^n b^n b^n) = a^n a^n b^n$. (Note that L_1 (resp. L_2) contains only one word of length $3n$, namely $a^n b^n b^n$ (resp. $a^n a^n b^n$).) Thus f can be written as $f = f_1 f_2 f_3$, $|f_1| = |f_2| = |f_3| = n$, where $f_1(a^n) = a^n$, $f_2(a^n) = b^n$, and $f_1(b^n) = b^n$. Hence $f_1 \in \{g_0, g_2\}^n$, $f_2 \in \{g_1, g_2\}^n$, $f_3 \in \{g_0, g_3\}^n$, i. e., $f \in \{g_0, g_2\}^n \{g_1, g_2\}^n \{g_0, g_3\}^n$. Since L_1 contains no word of length $3n + 1$ or $3n + 2$, $L_2^{L_1}$ contains all words in $\Sigma^{3n+1} \cup \Sigma^{3n+2}$. So we have

$$L_2^{L_1} = \bigcup_{n \geq 1} \left(\{g_0, g_2\}^n \{g_1, g_2\}^n \{g_0, g_3\}^n \cup \Sigma^{3n+1} \cup \Sigma^{3n+2} \right).$$

Since

$$L_2^{L_1} \cap (\{g_0, g_1\}^3)^* = \{g_0^n g_1^n g_0^n \mid n \geq 1\}$$

we infer that $L_2^{L_1}$ is not context-free. □

Now we turn to context-sensitive languages.

Theorem 3.2 *Let T_1 and T_2 be deterministic linear bounded automata accepting $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, respectively. Let $\Sigma = \Sigma_2^{\Sigma_1}$. Then there exists a deterministic linear bounded automaton T accepting $L_2^{L_1} \subseteq \Sigma^*$.*

Proof. Without loss of generality we assume that T_1 and T_2 hold on every input word. T works as follows: The tape of T is partitioned in three traces. Trace 1 contains the input word $f \in \Sigma^*$. On trace 3 the words $w \in \Sigma_1^{|f|}$ are generated in lexicographical order. For each $w \in \Sigma_1^{|f|}$, T checks whether $w \in L_1$ implies $f(w) \in L_2$. For that purpose, T copies w from trace 3 to trace 2 and then simulates T_1 on w. If T_1 does not accept w, i. e., $w \notin L_1$, this implication is true and T generates and checks the next word $w \in \Sigma_1^{|f|}$. If T_1 accepts w, i. e., $w \in L_1$, then $f(w)$ is computed on trace 2 (from f on trace 1 and w on trace 3). Then T_2 is simulated on $f(w)$ on trace 2. If T_2 does not accept $f(w)$, i. e., $f(w) \notin L_2$, then T stops without accepting f. If T_2 accepts $f(w)$, i. e., $f(w) \in L_2$, T generates and checks the next word $w \in \Sigma_1^{|f|}$.

If, in this way, it turns out that $w \in L_1$ implies $f(w) \in L_2$ for all $w \in \Sigma_1^*$ then the input word f is accepted.

Theorem 3.3 *Let T_1 and T_2 be nondeterministic linear bounded automata accepting $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, respectively. Let $\Sigma = \Sigma_2^{\Sigma_1}$ Then there exists a nondeterministic linear bounded automaton T accepting $L_2^{L_1} \subseteq \Sigma^*$.*

Proof. The construction of T is similar to the construction of the previous theorem, but some more care is necessary: Due to the nondeterminism, it might happen that for $w \in L_1$ the simulation of T_1 on w terminates in a nonaccepting state making T "believe" that $w \notin L_1$. To overcome these difficulties, we use the result of Immermann [8] and Szelepcsényi [10] that the family of context-sensitive languages is closed under complementation. Let U_1 be a nondeterministic linear bounded automaton accepting $\Sigma_1^* - L_1$. As in the proof of the previous theorem we assume that T_1, T_2, and U_1 hold on every input word.

Now modify the construction of T in the previous theorem as follows: If the simulation of T_1 on w terminates in an nonaccepting state, U_1 is simulated on w. If U_1 accepts w, then clearly $w \notin L_1$ and T can turn to the next word in $\Sigma^{|f|}$. Otherwise T stops without accepting f since, due to the nondeterminism, a decision whether or not $w \in L_1$ cannot be made.

Regarding the simulation of T_2 on $f(w)$, no modification is necessary.

Corollary 3.1 *Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ be context-sensitive languages and $\Sigma = \Sigma_2^{\Sigma_1}$. Then $L_2^{L_1} \subseteq \Sigma^*$ is again a context-sensitive language.*

The next two theorems can be proven in an analogous manner.

Theorem 3.4 *Let $S(n) \geq n$ be a measurable space function. If L_1 and L_2 are accepted by a deterministic (resp. nondeterministic) Turing machine of tape complexity $S(n)$, then $L_2^{L_1}$ is accepted by a deterministic (resp. nondeterministic) Turing machine of tape complexity $S(n)$.*

Proof. Since $S(n)$ is measurable we can assume that T_1, T_2, and U_1 hold for every input (see Harrison [6]). Furthermore, the result of Immermann [8] and Szelepcsényi [10] is valid also for $S(n) \geq n$.

Theorem 3.5 *If L_1 and L_2 are recursive languages, then so is $L_2^{L_1}$.*

Proof. The Turing machines T_1 and T_2 are deterministic and hold on every input.

Theorem 3.6 *Let $\mathfrak{F}$ be one of the families of languages considered in Theorems 3.1–3.5. Then $(\mathcal{L}_{\mathfrak{F}}; \leq, +, \times)$ is a Heyting algebra where the class $\emptyset$ is the 0-element. Hence, items (1)–(14) of Corollary 2.3 are valid for $\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3 \in \mathcal{L}_{\mathfrak{F}}$.*

4 Meet and join irreducible languages

Let $L \in \Sigma^*$. The language $C \subseteq L$ is a *core* of L if $L \sim C$ and $L \not\sim L_1$ for any proper subset L_1 of C; or equivalently, if (i) $L \sim C$ and (ii) $L_1 \subseteq C$, $L_1 \sim L$ imply $L_1 = C$.

Lemma 4.1 *Every language L has a core.*

Proof. Let $L \subseteq \Sigma^*$ and let $\Sigma_1 \subseteq \Sigma$ be an alphabet with the least number of elements so that $L \sim L \cap \Sigma_1^* := C$. We will prove that C is a core of L.

Let $L_1 \sim L$ be a subset of C. Then there is a morphism f of C into L_1. The morphism f induces a permutation π of Σ_1 because f is length preserving and because of the minimality of C. It follows that f is a one-to-one map of C into C. Hence f is an automorphism of C because there are only finitely many words in C of any given length n which implies $C = L_1$.

Lemma 4.2 *Let $L \subseteq \Sigma^*$ and $C = L \cap \Sigma_1^*$ be a core of L, where $\Sigma_1 \subseteq \Sigma$ is an alphabet given in the proof of Lemma 4.1. Let $f : \Sigma^* \to \Sigma^*$ be a morphism with $f(\Sigma_1) \subseteq \Sigma_1$. If $f(L) \subseteq L$ then $f : \Sigma_1^* \to \Sigma_1^*$ is an automorphism such that $f(C) = C$.*

Proof. Assume that $f(\Sigma_1) = \Sigma_2 \subsetneq \Sigma_1$. Then we claim that $L \sim L \cap \Sigma_2^*$, a contradiction to the minimality of $|\Sigma_1|$.

(i) Since $L \cap \Sigma_2^* \subseteq L$ we obtain $L \cap \Sigma_2^* \leq L$. (ii) Since $C \sim L$ there exists a morphism g such that $g(L) \subseteq C$. This implies $f(g(L)) \subseteq f(C) \subseteq f(L) \cap \Sigma_2^* \subseteq L \cap \Sigma_2^*$. Hence, $L \leq L \cap \Sigma_2^*$ and our claim is proven.

Since $f(C) \subseteq f(L) \cap \Sigma_1^* \subseteq L \cap \Sigma_1^* = C$ and f is an automorphism, we infer $f(C) = C$.

Lemma 4.3 *If $L_1 \sim L_2$ and C_1 is a core of L_1 and C_2 is a core of L_2 then C_1 and C_2 are isomorphic. Hence, any two cores of a language are isomorphic.*

Proof. Let $C_1 = L_1 \cap \Sigma_1^*$ and $C_2 = L_2 \cap \Sigma_2^*$, where Σ_1 and Σ_2 have least numbers of elements as in the proof of Lemma 4.1. Since $L_1 \sim L_2$ there exist morphisms f and g such that $f(L_1) \subseteq L_2$, $g(L_2) \subseteq L_1$. Since $(g \circ f)(L_1) \subseteq L_1$ and $(f \circ g)(L_2) \subseteq L_2$, Lemma 4.2 implies that $g \circ f : \Sigma_1^* \to \Sigma_1^*$ and $f \circ g : \Sigma_2^* \to \Sigma_2^*$ are automorphisms with $(g \circ f)(C_1) = C_1$ and $(f \circ g)(C_2) = C_2$. Hence, $f : \Sigma_1^* \to \Sigma_2^*$ and $g : \Sigma_2^* \to \Sigma_1^*$ are isomorphisms.

A family $\mathfrak{F}$ of languages is *stable* if:

1. $\mathfrak{F}$ is a lattice family.
2. $\mathfrak{F}$ is closed under exponentiation.
3. $\mathfrak{F}$ is closed under intersection with regular languages.

Note that by 3. the core of any element L in $\mathfrak{F}$ is an element of $\mathfrak{F}$. Observe further that the set of $\sim$-equivalence classes of every stable family of languages forms a Heyting algebra.

Theorem 4.1 *The following families of languages are stable:*

(i) The family of regular languages.
(ii) The family of languages accepted by deterministic linear bounded automata.
(iii) The family of context-sensitive languages.

(iv) The family of recursive languages.
(v) The family of languages accepted by deterministic Turing machines of tape complexity $S(n)$, where $S(n) \geq n$ is measurable.
(vi) The family of languages accepted by nondeterministic Turing machines of tape complexity $S(n)$, where $S(n) \geq n$ is measurable.

Proof. All the families of languages (i)–(iv) are lattice families by Theorem 2.1 and are closed under exponentiation by Theorems 3.1, 3.2, 3.3, 3.5. Moreover they are closed under intersection with regular languages by Ginsburg [4] (see page 10 for (i), page 13 for (ii) and (iii), page 9 for (iv)).

Clearly, the families of languages (v) and (vi) are lattice families and they are closed under exponentiation by Theorem 3.4. Moreover, they are clearly closed under intersection with regular languages.

Lemma 4.4 *If $\mathfrak{F}$ is a stable family of languages and $L \in \mathfrak{F}$ and $L = L_1 + L_2$ then $L_1 \in \mathfrak{F}$ and $L_2 \in \mathfrak{F}$.*

Proof. If $L = L_1 + L_2$ with $L_1 \in \Sigma_1^*$ and $L_2 \in \Sigma_2^*$ and $\Sigma_1 \cap \Sigma_2 = \emptyset$ then $L \cap \Sigma_1^* = L_1$ and $L \cap \Sigma_2^* = L_2$. Hence L_1 and L_2 are again in $\mathfrak{F}$.

In order to investigate the structure of the lattice $\mathcal{L}_{\mathfrak{F}}$ we attempt to determine the join and meet irreducible elements of $\mathcal{L}_{\mathfrak{F}}$.

The element $\mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$ is *join irreducible in* $\mathcal{L}_{\mathfrak{F}}$ if $\mathrm{L} = \mathrm{L}_1 + \mathrm{L}_2$ with $\mathrm{L}_1 \in \mathcal{L}_{\mathfrak{F}}$ and $\mathrm{L}_2 \in \mathcal{L}_{\mathfrak{F}}$ implies $\mathrm{L} = \mathrm{L}_1$ or $\mathrm{L} = \mathrm{L}_2$.

A language L is *coherent in* $\mathfrak{F}$ if for all languages $L_1 \in \mathfrak{F}$ and $L_2 \in \mathfrak{F}$ the equation $L = L_1 + L_2$ implies that $L_1 = \emptyset$ or $L_2 = \emptyset$. According to Lemma 4.4, the language L is coherent in $\mathfrak{F}$ if for all languages L_1 and L_2 the equation $L = L_1 + L_2$ implies that L_1 or L_2 is empty, that is if L is coherent in the set of all languages. In which case we call L coherent.

Let L be a language over the alphabet Σ. For $a, b \in \Sigma$ we write $a \equiv b$ if there is a sequence $a = a_1, a_2, \ldots, a_n = b$ of elements of Σ, so that for all $1 \leq i \leq n-1$ the letters a_i and a_{i+1} are together letters in some word of L. It follows easily that $\equiv$ is an equivalence relation on Σ.

Let L be a language over the alphabet Σ and let $\Sigma_1, \Sigma_2, \ldots, \Sigma_n$ be the $\equiv$-equivalence classes of Σ. Let $L_i = \Sigma_i^* \cap L$, $1 \leq i \leq n$. We write $w \equiv v$ for two words $w, v \in L$ if for some L_i both words w and v are in L_i. Clearly $\equiv$ is an equivalence relation on L.

Lemma 4.5 *Let L be a language and $L_1, L_2, \ldots, L_n$ be the $\equiv$-equivalence classes of L. Then $L = L_1 + L_2 + \cdots + L_n$.*

Proof. Obvious.

Corollary 4.1 *Let $\mathfrak{F}$ be a stable family of languages. A language $L \in \mathfrak{F}$ is coherent if and only if L has only one $\equiv$-equivalence class.*

Observe that if L is coherent and L_1 isomorphic to L then L_1 is coherent.

Let $\mathfrak{F}$ be a stable family of languages. It follows from Lemma 4.2 that if $\mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$ and $L_1 \in \mathrm{L}$ and $L_2 \in \mathrm{L}$ then the cores of L_1 and L_2 are isomorphic. We define the *core of* L to be the core of any of its elements. The core of L is uniquely determined up to isomorphism and is an element of $\mathfrak{F}$ and hence an element of L. A language $C \in \mathfrak{F}$ is a *core*, or a core in $\mathfrak{F}$, if it is the core of the $\sim$-equivalence class of $\mathcal{L}_{\mathfrak{F}}$ containing C. Observe that if C is a core and L is a proper subset of C then $C \not\sim L$.

Lemma 4.6 *Let $\mathfrak{F}$ be a stable family of languages and $C \in \mathfrak{F}$. Let $L_1, L_2, \ldots, L_n$ be the $\equiv$-equivalence classes of C. Then C is a core if and only if L_i is a core for all $1 \leq i \leq n$ and if $L_i \not\leq L_j$ whenever $i \neq j$.*

Proof. (i) Let C be a core and assume for a contradiction that, say L_1, is not a core. Then there is a proper subset L_1' of L_1 with $L_1' \sim L_1$. It follows that $C' := L_1' + L_2 + \cdots + L_n$ is a proper subset of C with $C' \sim C$.

If, say $L_1 \leq L_2$, then $C' := L_2 + L_3 + \cdots + L_n$ is a proper subset of C with $C' \sim C$.

(ii) Let, for all $1 \leq i \leq n$, the language L_i be a core and $L_i \not\leq L_j$ whenever $i \neq j$ and let C' be a subset of C with $C \sim C'$. Let $L_i' := L_i \cap C'$. Then $C' = L_1' + L_2' + \cdots + L_n'$. If C' is a proper subset of C then L_k' is a proper subset of L_k for some k with $1 \leq k \leq n$. Because L_k is a core there is no morphism of L_k to L_k'.

On the other hand there is a morphism f of C into C' because $C \sim C'$. The morphism f maps L_i into L_i for every $1 \leq i \leq n$ because if $w \equiv v$ then $f(w) \equiv f(v)$ and $L_i \not\leq L_j$ if $i \neq j$. We arrive at a contradiction because the restriction of f to L_j maps L_j into L_j' for $1 \leq j \leq n$.

Theorem 4.2 *Let $\mathfrak{F}$ be a stable family of languages. Then $\mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$ is join irreducible if and only if the core of L is coherent.*

Proof. (i) Let $\mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$ be join irreducible and $L \in \mathrm{L}$ and C be a core of L. Then $C \in \mathrm{L}$ because $\mathfrak{F}$ is stable. Assume for a contradiction that C is not coherent. Then, according to Lemma 4.4, there are languages $L_1 \in \mathfrak{F}$ and $L_2 \in \mathfrak{F}$ with $C = L_1 + L_2$ and $L_1 \neq \emptyset \neq L_2$. Let L_1 be the $\sim$-equivalence class of $\mathcal{L}_{\mathfrak{F}}$ containing L_1 and L_2 be the $\sim$-equivalence class of $\mathcal{L}_{\mathfrak{F}}$ containing L_2. Then $\mathrm{L} = \mathrm{L}_1 + \mathrm{L}_2$.

But $\mathrm{L} = \mathrm{L}_1 + \mathrm{L}_2$ is a contradiction to L being join irreducible in $\mathcal{L}_{\mathfrak{F}}$ because if $\mathrm{L} = \mathrm{L}_1$ then $C \sim L_1$. Which implies $L_2 = \emptyset$ because of the minimality of C.

(ii) Let the core C of L be coherent. Assume for a contradiction that L is not join irreducible. Then there are elements L_1 and L_2 in $\mathcal{L}_{\mathfrak{F}}$ with $\mathrm{L} = \mathrm{L}_1 + \mathrm{L}_2$ and $\mathrm{L}_1 \neq \mathrm{L} \neq \mathrm{L}_2$.

Let C_1 be a core of L_1 and C_2 be a core of L_2. We obtain $C \sim C_1 + C_2$. Let $C' = C_1 + C_2$. Let $C_1 = L_{1,1} + L_{1,2} + \cdots + L_{1,n}$ and $C_2 = L_{2,1} + L_{2,2} + \cdots + L_{2,m}$ be the partitions of C_1 and C_2 respectively into $\equiv$-equivalence classes. Because C_1 and C_2 are cores it follows, according to Lemma 4.6, that all those languages $L_{i,j}$ are cores and all the languages of the form $L_{1,j}$ are pairwise incomparable under $\leq$ and all the languages of the form $L_{2,j}$ are pairwise incomparable under $\leq$.

Let I be the set of all numbers $1 \leq i \leq n$ for which there is no $1 \leq j \leq m$ so that $L_{1,i} \leq L_{2,j}$. Let J be the set of all numbers $1 \leq j \leq m$ for which there is no $1 \leq i \leq n$ so that $L_{2,j} \leq L_{1,i}$. Observe that $I \neq \emptyset$ because otherwise $C_1 \leq C_2$ and hence $\mathrm{L}_1 \leq \mathrm{L}_2$ and hence $\mathrm{L} \sim \mathrm{L}_2$. Similarly $J \neq \emptyset$. Let $D_1 = \Sigma_{i \in I} L_{1,i}$ and $D_2 = \Sigma_{j \in J} L_{2,j}$ and $D = D_1 + D_2$. Then $D \sim C'$ and $D_1 \neq \emptyset$ and $D_2 \neq \emptyset$ and D is not coherent and D is a core according to Lemma 4.6.

There is a morphism g of C into D and a morphism f of D into C. Then $f \circ g$ is a morphism of C into C which is by Lemma 4.3 an automorphism of C. Similarly $g \circ f$ is an automorphism of D. If follows that C and D are isomorphic and hence that C is not coherent.

It follows that if $\mathfrak{F}$ is a stable family of languages then there is an efficient algorithm to determine whether $\mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$ is join irreducible for some given language $L \in \mathfrak{F}$.

Let $\mathfrak{F}$ be a stable family of languages. The equivalence class $\mathrm{L} \in \mathcal{L}_{\mathfrak{F}}$ is *meet irreducible in the lattice* $\mathcal{L}_{\mathfrak{F}}$ if $\mathrm{L} = \mathrm{L}_1 \times \mathrm{L}_2$ with L_1 and L_2 in $\mathcal{L}_{\mathfrak{F}}$ implies $\mathrm{L} = \mathrm{L}_1$ or $\mathrm{L} = \mathrm{L}_2$. Observe that L is meet irreducible if and only if $L \sim L_1 \times L_2$ with $L_1, L_2 \in \mathfrak{F}$ implies $L \sim L_1$ or $L \sim L_2$ for every $L \in \mathrm{L}$. In which case we say that L is *meet irreducible in* $\mathfrak{F}$. It follows that L is meet irreducible in $\mathcal{L}_{\mathfrak{F}}$ if and only if every language $L \in \mathrm{L}$ is meet irreducible in $\mathfrak{F}$.

Lemma 4.7 *A language $L \in \mathfrak{F}$ is meet irreducible in a stable family $\mathfrak{F}$ of languages if and only if $L_1 \not\leq L$ implies $L^{L_1} \sim L$ for all languages $L_1 \in \mathfrak{F}$.*

Proof. (i) Let L be meet irreducible in $\mathfrak{F}$ and $L_1 \in \mathfrak{F}$ with $L_1 \not\leq L$. Then

$$(L + L_1) \times L^{L_1} \sim L \times L^{L_1} + L_1 \times L^{L_1} \leq L + L \sim L$$

by Theorem 2.3 item (4). and Lemma 2.6. Because $L + L_1 \not\sim L$ it follows that $L \sim L^{L_1}$.

(ii) Assume $L_1 \not\leq L$ implies $L \sim L^{L_1}$ for all languages $L_1 \in \mathfrak{F}$. Let $L_1 \times L_2 \sim L$. Then $L_2 \leq L^{L_1} \sim L$ according to Lemma 2.7. It follows that $L_2 \leq L$. If $L_2 < L$ then $L_1 \times L_2 < L$, hence $L_2 \sim L$.

The morphism class $\overset{\circ}{1}$ is trivially meet irreducible in every stable family $\mathfrak{F}$. Hence a language L is meet irreducible if there is a morphism h of $\{a\}^*$ into

L. The empty language is not meet irreducible in $\mathfrak{F}$ if $\mathfrak{F}$ contains two languages $L_1 \neq \emptyset$ and $L_2 \neq \emptyset$ so that for every $n \in \omega$ if L_1 contains a word of length n then L_2 does not contain a word of length n. It follows that $L_1 \times L_2 = \emptyset$. If there is an $n \in \omega$ so that every language $L \in \mathfrak{F}$ contains only words of length n then the language $\emptyset$ is meet irreducible in $\mathfrak{F}$.

Lemma 4.8 *Let $\mathfrak{F}$ be a stable family of languages and $L \in \mathfrak{F}$. If there are two languages L_1 and L_2 in $\mathfrak{F}$ so that $L_1 \not\leq L$ and $L_2 \not\leq L$ and $L_1 \times L_2 \leq L$ then L is not meet irreducible in $\mathfrak{F}$.*

Proof. Because $\mathfrak{F}$ is stable, it is closed under $+$ and hence $L + L_1$ and $L + L_2$ are elements of $\mathfrak{F}$. We calculate:

$$(L + L_1) \times (L + L_2) \sim L \times L + L \times L_2 + L \times L_1 + L_1 \times L_2 \sim L.$$

For $a \in \Sigma_\infty$ and $n \in \omega$ let a^n be the word $aa \ldots a$ of length n. (a^0 being the empty word ε.)

Lemma 4.9 *Let $\mathfrak{F}$ be a stable family of languages and $L \in \mathfrak{F}$ and let $n, m \in \omega$ with $n \neq m$. If there is no $a \in \Sigma_\infty$ with $a^n \in L$ and if there is no $a \in \Sigma_\infty$ with $a^m \in L$ then L is not meet irreducible in $\mathfrak{F}$.*

Proof. Let $b \in \Sigma_\infty$. It follows that $\{b^n\} \not\leq L$ and that $\{b^m\} \not\leq L$. Because $\mathfrak{F}$ is stable it contains the languages $\{b^n\}$ and $\{b^m\}$. The language $\{b^n\} \times \{b^m\}$ is empty which implies $\{b^n\} \times \{b^m\} \leq L$. The lemma follows now from Lemma 4.8.

Lemma 4.10 *Let $\mathfrak{F}$ be a stable family of languages and $L \in \mathfrak{F}$ and let n_1, n_2, m_1, m_2 be four pairwise different elements of ω.*

If there is no $a \in \Sigma_\infty$ so that $a^{n_1} \in L$ and $a^{n_2} \in L$ and if there is no $a \in \Sigma_\infty$ so that $a^{m_1} \in L$ and $a^{m_2} \in L$ then L is not meet irreducible in $\mathfrak{F}$.

Proof. Let $b \in \Sigma_\infty$. It follows that $\{b^{n_1}, b^{n_2}\} \not\leq L$ and that $\{b^{m_1}, b^{m_2}\} \not\leq L$ and that the languages $\{b^{n_1}, b^{n_2}\}$ and $\{b^{m_1}, b^{m_2}\}$ are elements of $\mathfrak{F}$. Then $\{b^{n_1}, b^{n_2}\} \times \{b^{m_1}, b^{m_2}\} = \emptyset \leq L$ and the lemma follows from Lemma 4.8.

Corollary 4.2 *Let $\mathfrak{F}$ be a stable family of languages and $L \in \mathfrak{F}$. If L is meet irreducible in $\mathfrak{F}$ then there exists $n \in \omega$ and $a \in \Sigma_\infty$ so that $a^m \in L$ for all $m \in \omega$ with $n \neq m$.*

Proof. Follows from Lemma 4.9 and Lemma 4.10

Let $S \subseteq \omega$. Then we define the family $\mathfrak{F}_S$ of languages by the following condition: $L \in \mathfrak{F}_S$ iff, for all $w \in L$, $|w| \in S$. Moreover, let L be a language over Σ. Then we denote

$$L^S := L \cap \bigcup_{n \in S} \Sigma^n .$$

For $m \in \omega$ we denote by $\underline{m}$ the set $\{0, 1, 2, \ldots, m-1\}$.

Lemma 4.11 *Let K and L be languages so that $K^{\underline{m}} \leq L$ for every $m \in \omega$. Then $K \leq L$.*

Proof. For every $m \in \omega$ let H_m be the set of morphisms of $K^{\underline{m}}$ into L and let $H_\omega := \bigcup_{m \in \omega} H_m$. Let $f \leq h$ if $f \in H_l$ and $h \in H_m$ with $l \leq m$ and f is the restriction of h to $K^{\underline{l}}$.

It follows that $(H_\omega; \leq)$ is an infinite tree in which every element has finitely many successors. Hence Königs Tree Lemma applies and we obtain a morphism of K into L.

Theorem 4.3 *Let $\mathfrak{F}$ be a stable family of languages and $L \in \mathfrak{F}$. The language L is meet irreducible in $\mathfrak{F}$ if and only if there is $n \in \omega$ and $a \in \Sigma_\infty$ so that $a^m \in L$ for all $m \neq n$ and if the languages $L^{\underline{m}}$ are meet irreducible in $\mathfrak{F}_{\underline{m}}$ for all $m \in \omega$.*

Proof. (i) Let L be meet irreducible and $m \in \omega$. Because of Corollary 4.2 the only thing left to prove is that the language $L^{\underline{m}}$ is meet irreducible in $\mathfrak{F}_{\underline{m}}$. Assume not. Then there are two languages L_1 and L_2 in $\mathfrak{F}_{\underline{m}}$ with $L_1 \not\leq L^{\underline{m}}$ and $L_2 \not\leq L^{\underline{m}}$ so that $L_1 \times L_2 \sim L^{\underline{m}}$. It follows that $L_1 \not\leq L$ and $L_2 \not\leq L$ and that $L_1 \times L_2 \leq L$. Because L_1 and L_2 are both finite languages, L_1 and L_2 are elements of $\mathfrak{F}$. Using Lemma 4.8 we arrive at a contradiction to the assumption that L is meet irreducible.

(ii) Let, for every $m \in \omega$ the language $L^{\underline{m}}$ be meet irreducible in $\mathfrak{F}_{\underline{m}}$ and $a \in \Sigma_\infty$ so that $a^m \in L$ for all $m \neq n$. Assume for a contradiction that there are languages L_1 and L_2 in $\mathfrak{F}$ and with $L_1 \not\leq L$ and $L_2 \not\leq L$ so that $L_1 \times L_2 \sim L$.

It follows from Lemma 4.11 that there are k and l in ω so that $L_1^{\underline{k}} \not\leq L$ and $L_2^{\underline{l}} \not\leq L$. Let m be the maximum of k and l. Then $L_1^{\underline{m}} \not\leq L$ and $L_2^{\underline{m}} \not\leq L$. It follows from $L_1 \times L_2 \leq L$ that $L_1^{\underline{m}} \times L_2^{\underline{m}} \leq L^{\underline{m}}$ in contradiction to the assumption that for every $m \in \omega$ the language $L^{\underline{m}}$ is meet irreducible in $\mathfrak{F}_{\underline{m}}$.

References

1. Balbes R., Dwinger P.: Distributive Lattices. University of Missouri Press, Columbia, Missouri 65201, 1974.
2. Birkhoff G.: Extended arithmetic. Duke Math. J. 3(1937) 311–316.
3. Birkhoff G.: Generalized arithmetic. Duke Math. J. 12(1942) 283–302.
4. Ginsburg, S.: Algebraic and Automata-Theoretic Properties of Formal Languages. North-Holland, 1975.
5. Goldblatt R.: Topoi; The Categorial Analysis of Logic. Studies in Logic and the Foundations of Mathematics, 98, North-Holland.
6. Harrison, M. A.: Introduction to Formal Language Theory. Addison-Wesley, 1978.
7. Heyting A.: Die formalen Regeln der intuitionistischen Logik. Sitzungsberichte der Preußischen Akademie der Wissenschaften, Phys.-mathem. Klasse (1930) 42–56.
8. Immermann N.: NSPACE is closed under complement. SIAM Journal on Computing 17(1988) 935–938.
9. Sauer N.: Hedetniemi's Conjecture—a survey. Combinatorics, graph theory, algorithms and applications. Discrete Math. 229(2001), no. 1-3, 261–292.
10. Szelepcsényi R.: The method of forced enumeration for nondeterministic automata. Acta Informatica 26(1988) 279–284.