The dual of concatenation [1]

Alexander Okhotin

Department of Mathematics, University of Turku, Turku FIN–20014, Finland

---

Abstract

A binary language-theoretic operation is proposed, which is dual to the concatenation of languages in the same sense as the universal quantifier in logic is dual to the existential quantifier; the dual of Kleene star is defined accordingly. These operations arise whenever concatenation or star appear in the scope of negation. The basic properties of the new operations are determined in the paper. Their use in regular expressions and in language equations is considered, and it is shown that they often eliminate the need of using negation, at the same time having an important technical advantage of being monotone. A generalization of context-free grammars featuring dual concatenation is introduced and proved to be equivalent to the recently studied Boolean grammars.

Key words: Formal languages, semiring, regular expressions, language equations, deductive parsing, co-context-free languages, conjunctive grammars, Boolean grammars.

---

1 Introduction

The dual of a logical proposition $f(x_1, \ldots, x_n)$ is its transform under negation, $\neg f(\neg x_1, \ldots, \neg x_n)$. For instance, conjunction is the dual of disjunction, as stated by de Morgan's law. The existential and the universal quantifier are dual to each other, since $\neg(\exists x)P(x)$ is equivalent to $(\forall x)\neg P(x)$. In temporal logic, the operators F ("eventually") and G ("always") demonstrate the same kind of duality. The possibility of saying "eventually X" instead of the cumbersome "not always not X" is certainly helpful both for intuitive clarity and due to the technical difficulties associated with nonmonotone negation.

Email address: `okhotin@cs.utu.fi` (Alexander Okhotin).

[1] This work was carried out during the author's studies at Queen's University (Kingston, Ontario, Canada). A preliminary version of this paper was presented at the MFCS 2004 symposium held in Prague, Czech Republic, August 22–27, 2004, and its extended abstract appeared in: J. Fiala, V. Koubek, J. Kratochvíl (Eds.), *Mathematical Foundations of Computer Science*, LNCS 3153, Springer-Verlag, 2004, 698–710.

Among the commonly used language-theoretic operations, union and intersection are dual to each other, while concatenation is apparently without a dual. As a result, reasoning about language constructs like $\overline{\overline{L_1} \cdot \overline{L_2}}$ becomes as inconvenient as it would be to deal with the statement "not always not X", being forbidden to use the notion "eventually". This makes a significant contribution to the general attitude to complement in formal language theory as a "hard" operation.

It is not hard to define the operation dual to concatenation: recalling that concatenation of two languages is

$$L_1 \cdot L_2 = \{w \mid \text{there exists a factorization } w = uv\text{, such that } u \in L_1 \text{ and } v \in L_2\},$$

one can formally invert the predicate by replacing the existential quantifier with the universal quantifier and conjunction with disjunction:

$$L_1 \odot L_2 = \{w \mid \text{for every factorization } w = uv \text{ it holds that } u \in L_1 \text{ or } v \in L_2\}.$$

This is a binary operation on languages. Although it might look artificial at the first glance, one should bear in mind that such an operation invariably arises whenever concatenation occurs in the scope of negation. This gives a clear motivation for conducting a study of this operation, named *dual concatenation*.

In Sections 2 and 3, dual concatenation and its iterative counterpart, *dual star,* are formally introduced and their basic properties are established. Section 4 obtains closure/nonclosure results with respect to these operations for the most common families of languages. Sections 5 and 6 examine the use of dual concatenation together with or instead of concatenation in regular expressions and in language equations. A generalization of context-free grammars featuring explicit dual concatenation is defined in Section 7; it is shown to be equivalent to the recently studied Boolean grammars [12], which contain explicit negation. The final Section 8 summarizes the contributions of dual concatenation to these areas and argues for its importance.

2 The dual concatenation

Let us start from giving two equivalent definitions of the new operation.

Definition 1 *The dual concatenation of two languages* $L_1, L_2 \subseteq \Sigma^*$ *is defined as*

$$L_1 \odot L_2 = \{w \mid \textit{for every factorization } w = uv \textit{ it holds that } u \in L_1 \textit{ or } v \in L_2\}$$

Definition 2 *The dual concatenation of two languages* $L_1, L_2 \subseteq \Sigma^*$ *is defined as*

$$L_1 \odot L_2 = \overline{\overline{L_1} \cdot \overline{L_2}} \tag{1}$$

Theorem 1 *Definitions 1 and 2 are equivalent.*

PROOF. Writing down a formal negation of Definition 1, a string w is *not* in the dual concatenation of L_1 and L_2 according to that definition if and only if there exists a factorization $w = uv$, such that $u \notin L_1$ and $v \notin L_2$. This is in turn equivalent to $w \in \overline{L_1} \cdot \overline{L_2}$, which holds if and only if w is *not* in the language (1). $\square$

Theorem 2 (Algebraic properties of dual concatenation) *(1) Dual concatenation is associative, i.e., $L_1 \odot (L_2 \odot L_3) = (L_1 \odot L_2) \odot L_3$.*
(2) Dual concatenation is not commutative, i.e., $L_1 \odot L_2$ is not necessarily equal to $L_2 \odot L_1$.
(3) Σ^+ is a two-sided identity for dual concatenation, i.e., $L \odot \Sigma^+ = \Sigma^+ \odot L = L$.
(4) Σ^* is a two-sided zero for dual concatenation, i.e., $L \odot \Sigma^* = \Sigma^* \odot L = \Sigma^*$.
(5) Dual concatenation is distributive over intersection (including infinite intersection), i.e., $L \odot \bigcap_{i \in I} L_i = \bigcap_{i \in I}(L \odot L_i)$ for any index set I and $L, L_i \subseteq \Sigma^*$.

PROOF. For simplicity, all the proofs are based on the representation $L_1 \odot L_2 = \overline{\overline{L_1} \cdot \overline{L_2}}$ and the universally known corresponding results for concatenation.

Associativity: $L_1 \odot (L_2 \odot L_3) = \overline{\overline{L_1} \cdot \overline{L_2 \odot L_3}} = \overline{\overline{L_1} \cdot (\overline{L_2} \cdot \overline{L_3})} = \overline{(\overline{L_1} \cdot \overline{L_2}) \cdot \overline{L_3}} = \overline{\overline{L_1 \odot L_2} \cdot \overline{L_3}} = (L_1 \odot L_2) \odot L_3$.

Noncommutativity: Suppose dual concatenation is commutative. Then, for every $L_1, L_2 \in \Sigma^*$, $L_1 \cdot L_2 = \overline{\overline{L_1} \odot \overline{L_2}} = \overline{\overline{L_2} \odot \overline{L_1}} = L_2 \cdot L_1$, and hence concatenation is also commutative, which is known to be untrue.

Identity: for every $L \subseteq \Sigma^*$, $L \odot \Sigma^+ = \overline{\overline{L} \cdot \overline{\Sigma^+}} = \overline{\overline{L} \cdot \{\varepsilon\}} = \overline{\overline{L}} = L$. Similarly it is proved that $\Sigma^+ \odot L = L$ for all L.

Zero: $L \odot \Sigma^* = \overline{\overline{L} \cdot \overline{\Sigma^*}} = \overline{\overline{L} \cdot \varnothing} = \overline{\varnothing} = \Sigma^*$. Similarly, $\Sigma^* \odot L = \Sigma^*$ for all L.

Distributivity:

$$L \odot \bigcap_{i \in I} L_i = \overline{\overline{L} \cdot \overline{\bigcap_{i \in I} L_i}} = \overline{\overline{L} \cdot \bigcup_{i \in I} \overline{L_i}} = \overline{\bigcup_{i \in I} (\overline{L} \cdot \overline{L_i})} = \bigcap_{i \in I} \overline{\overline{L} \cdot \overline{L_i}} = \bigcap_{i \in I} (L \odot L_i). \quad \square$$

Corollary 1 (The dual semiring of formal languages) *Formal languages over an alphabet Σ form a semiring with $\cap$ as sum and $\odot$ as product, with zero Σ^* and identity Σ^+, denoted $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+ \rangle$. This semiring is complete with respect to infinite intersections and naturally ordered by the relation of containment (see Kuich [6] for definitions).*

PROOF. The axioms of a semiring are stated in parts 1, 3, 4 and 5 (for two-element I) of Theorem 2. Completeness of this semiring follows from the obvious properties of intersection and from part 5 of Theorem 2 (this time for arbitrary I's). The partial order on languages defined as "$L_1 \sqsubseteq L_2$ if and only if $L_1 \supseteq L_2$" satisfies the axiom "$L_1 \sqsubseteq L_2$ if and only if $L_1 \cap L = L_2$ for some L", as it suffices to take $L = L_2$; hence it is a natural order for this semiring. $\square$

The semiring $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+\rangle$ is dual to the classical semiring of formal languages $\langle 2^{\Sigma^*}, \cup, \cdot, \varnothing, \{\varepsilon\}\rangle$ [6]; some of its further properties will be established later.

Theorem 3 (Analytic properties of dual concatenation) *(1) Dual concatenation is monotone with respect to inclusion, i.e., if $L_1 \subseteq L_1'$ and $L_2 \subseteq L_2'$, then $L_1 \odot L_2 \subseteq L_1' \odot L_2'$.*

(2) Dual concatenation is $\cup$- and $\cap$-continuous, i.e., for every two sequences of languages $\{L_n'\}_{n=1}^{\infty}$ and $\{L_n''\}_{n=1}^{\infty}$ that are both increasing (both decreasing, resp.), the sequence $\{L_n' \odot L_n''\}_{n=1}^{\infty}$ is also increasing (decreasing, resp.) and its limit equals $\sup_{n\geqslant 1}\{L_n'\} \odot \sup_{n\geqslant 1}\{L_n''\}$ ($\inf_{n\geqslant 1}\{L_n'\} \odot \inf_{n\geqslant 1}\{L_n''\}$, resp.).

PROOF. Monotonicity: Let $w \in L_1 \odot L_2$. Then for every factorization $w = u_1u_2$ there exists i, such that $u_i \in L_i$. Since $L_1 \subseteq L_1'$ and $L_2 \subseteq L_2'$, for every factorization $w = u_1u_2$ there exists i, such that $u_i \in L_i'$. This means $w \in L_1' \odot L_2'$.

Continuity: Let us prove $\cup$-continuity. If both $\{L_n'\}_{n=1}^{\infty}$ and $\{L_n''\}_{n=1}^{\infty}$ are increasing, then $\{L_n' \odot L_n''\}_{n=1}^{\infty}$ is increasing by the monotonicity of dual concatenation.

If $w \in \sup\{L_n' \odot L_n''\}$ then there exists $k \geqslant 1$, such that $w \in L_k' \odot L_k''$. Since $L_k' \subseteq \sup\{L_n'\}$ and $L_k'' \subseteq \sup\{L_n''\}$, $L_k' \odot L_k'' \subseteq \sup\{L_n'\} \odot \sup\{L_n''\}$ by the monotonicity of dual concatenation, and hence $w \in \sup\{L_n'\} \odot \sup\{L_n''\}$.

Conversely, if $w \in \sup\{L_n'\} \odot \sup\{L_n''\}$, then for every factorization $w = uv$, $u \in \sup\{L_n'\}$ or $v \in \sup\{L_n''\}$. By the definition of least upper bound, for every such $u_i \in \sup\{L_n'\}$ there exists $k_i \geqslant 1$, such that $u_i \in L_{k_i}'$, and similarly for every such $v_i \in \sup\{L_n''\}$ there is $\ell_i \geqslant 1$, for which $v_i \in L_{\ell_i}''$. Since w has finitely many substrings, the set $\{k_i, \ell_j\}$ is finite. Let $k = \max\{k_i, \ell_j\}$. By the monotonicity of $\{L_n'\}_{n=1}^{\infty}$ and $\{L_n''\}_{n=1}^{\infty}$, all the relevant substrings will be in L_k' and L_k'', and therefore $w \in L_k' \odot L_k'' \subseteq \sup\{L_n' \odot L_n''\}$. $\square$

An important case of concatenation is *linear concatenation*, where a singleton $\{a\}$ ($a \in \Sigma$) is left- or right- concatenated to a language: $a{\cdot}L$ or $L{\cdot}a$. What happens if the dual of concatenation is similarly restricted? Consider *linear dual concatenation* defined as $\overline{a} \odot L$ and $L \odot \overline{a}$. It turns out that linear concatenation and linear dual concatenation can be expressed through each other:

Theorem 4 *For every language* $L \subseteq \Sigma^*$ *and for every* $a \in \Sigma$, $\overline{a} \odot L = aL \cup \overline{a \cdot \Sigma^*}$ *and* $a \cdot L = \overline{a} \odot L \cap a \cdot \Sigma^*$.

Let us now show how a nontrivial language can be obtained from languages of a simple form using dual concatenation.

Example 1 *Define the following three languages:*

$$L_a = \{xay \mid x, y \in \{a,b\}^*,\ |x| = |y|\}, \tag{2a}$$

$$L_b = \{xby \mid x, y \in \{a,b\}^*,\ |x| = |y|\}, \tag{2b}$$

$$L_{even} = \{aa, ab, ba, bb\}^* \tag{2c}$$

Then $L = L_a \odot (L_b \cup L_{even}) \cap L_b \odot (L_a \cup L_{even}) = \{ww \mid w \in \{a,b\}^*\}$.

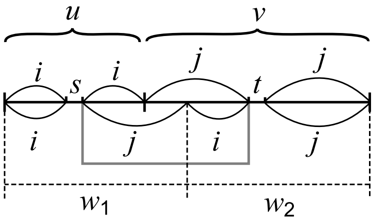

Fig. 1. How the language $\{ww \mid w \in \{a,b\}^*\}$ is obtained in Example 1.

Let us explain how this construct ensures that $x \in L$ if and only if $x = ww$ for some w. Consider a string x of even length. For every factorization of x into two strings of odd length, $x = uv$, let $s, t \in \{a,b\}$ be the middle symbols in u and v respectively, as shown in Figure 1. It is easy to see that s and t occupy the same relative position in the two halves of x, and hence it should be checked that $s = t$. The requirement that $x \in L_a \odot (L_b \cup L_{even})$ states that, for every such factorization, $s = a$ or $t = b$; so, if $s = b$, then t also has to be b. Similarly, $x \in L_b \odot (L_a \cup L_{even})$ ensures that if $s = a$, then t must equal a as well. The use of L_{even} allows to handle factorizations of x into even-length strings, as well as to enforce that the length of x is even.

3 The dual star

Kleene closure, $L^* = \bigcup_{n=0}^{\infty} L^n$, admits an equivalent representation as the set of all strings w, such that there exists a number $n \geqslant 0$ and a factorization $w = w_1 \ldots w_n$, such that for all i ($1 \leqslant i \leqslant n$), $w_i \in L$. Let us take a formal dual of this representation, thus obtaining a new language-theoretic operation. The properties of this operation will now be investigated.

Definition 3 *Let $L \subseteq \Sigma^*$ be a language. The dual star of L is defined as*

$$L^{\circledast} = \{w \mid \textit{for every } n \geqslant 0 \textit{ and for every factorization } w = w_1 \ldots w_n \\ \textit{there exists } i \textit{ (}1 \leqslant i \leqslant n\textit{), such that } w_i \in L\} \tag{3}$$

Definition 4 *Let $L \subseteq \Sigma^*$ be a language. Define the dual star of L as*

$$L^{\circledast} = \overline{\overline{L}^{*}} \tag{4}$$

Before proceeding to the proof of the equivalence of these two definitions, consider the third way of defining the dual star, by an iterative application of dual concatenation. Recalling the semiring $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+ \rangle$ with dual concatenation as the associative product, consider *powers* of an element defined as follows:

Definition 5 *The n-th dual concatenation power of a language L is defined as $L^{\textcircled{n}} = L \odot \ldots \odot L$, where L is repeated n times; if $n = 0$, assume $L^{\textcircled{n}} = \Sigma^+$.*

Definition 6 *The n-th dual concatenation power of L is defined as $L^{\textcircled{n}} = \{w | \textit{for every factorization } w = w_1 \ldots w_n \textit{ there exists } i \textit{ (}1 \leqslant i \leqslant n\textit{), such that } w_i \in L\}$.*

Theorem 5 *Definitions 5 and 6 are equivalent.*

PROOF. Induction on n.

Basis $n = 0$: According to Definition 6, $\varepsilon \notin L^{\textcircled{0}}$, since for the factorization of ε into zero substrings (as $\varepsilon = \varepsilon$, this is a valid factorization) there cannot exist i, such that $1 \leqslant i \leqslant 0$. On the other hand, every other string is in $L^{\textcircled{0}}$, because there are no factorizations of $w \neq \varepsilon$ into zero substrings, and hence everything that is supposed to hold for every such factorization is true. So, $L^{\textcircled{0}} = \Sigma^+$ under both definitions.

Induction step. $w \in L \odot L \odot \ldots \odot L$ ($n+1$ repetitions) if and only if for every factorization $w = uv$, $u \in L$ or $v \in L \odot \ldots \odot L$ (n repetitions). By the induction hypothesis, this is equivalent to: for every factorization $w = uv$, $u \in L$ or, for every factorization $v = v_1 \ldots v_n$, $v_1 \in L$ or ... or $v_n \in L$. Rewrite this as follows: for every factorization $w = uv_1 \ldots v_n$, $u \in L$ or $v_1 \in L$ or ... or $v_n \in L$. $\square$

Definition 7 *Let $L \subseteq \Sigma^*$ be a language. The dual star of L is defined as*

$$L^{\circledast} = \bigcap_{n=0}^{\infty} L^{\textcircled{n}} \tag{5}$$

Theorem 6 *Definitions 3, 4 and 7 are equivalent.*

PROOF. (Def. 3 ⇔ Def. 4) A string w is *not* in (3) if and only if there exists $n \geqslant 0$ and a factorization $w = w_1 \ldots w_n$, such that $w_i \notin L$ for every i. This is in turn equivalent to $w \in \overline{L}^*$, which holds if and only if w is *not* in the language $\overline{\overline{L}^*}$.

(Def. 3 $\Leftrightarrow$ Def. 7) w is in $L^{\circledast}$ according to Definition 3 if and only if for every $n \geqslant 0$ and for every factorization $w = w_1 \dots w_n$ there exists i ($1 \leqslant i \leqslant n$), such that $w_i \in L$. Using the notation of Definition 6, this can be equivalently rewritten as: for every $n \geqslant 0$, $w \in L^{\textcircled{n}}$. This holds if and only if w is in (5). $\square$

According to its third definition, dual star is an infinite intersection (5), which is an infinite "sum" in the complete semiring $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+ \rangle$. Thus dual star has the following role within this semiring:

Proposition 1 *$\circledast$ is an abstract star operation in the semiring $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+ \rangle$.*

Theorem 7 *For every language L, (1) $\varepsilon \notin L^{\circledast}$ (cf.: $\varepsilon \in L^*$); (2) $L^{\circledast} \subseteq L \subseteq L^*$; (3) $X = L_0^{\circledast}$ is the greatest solution of the equation $X = X \odot L_0 \cap \Sigma^+$.*

PROOF. Consider Definition 7: taking $n = 0$, we obtain the first claim, while $n = 1$ establishes the second claim. As for the third claim, it is a known fact that the least solution of the equation $Y = Y \cdot L \cup \{\varepsilon\}$ is L^*. Take the complement of both sides of the equation: $\overline{Y} = \overline{Y \cdot L \cup \{\varepsilon\}}$, or, equivalently, $\overline{Y} = \overline{Y} \odot \overline{L} \cap \Sigma^+$; this does not alter the set of solutions, and hence L^* is still the least solution of the latter equation. Substituting $X = \overline{Y}$, one obtains the equation $X = X \odot \overline{L} \cap \Sigma^+$ with the greatest solution $\overline{L^*}$, which equals $\overline{L}^{\circledast}$. Setting $L = \overline{L_0}$ for a given L_0 proves the claim. $\square$

Example 2 *Consider the languages L_a, L_b, L_{even} from Example 1 and define linear context-free languages*

$$L_1 = c \cdot L_a \cup L_b \cdot d \cup L_{even} \cdot d \cup c \cdot L_{even} \cdot d \cup \{a, b\}^* \quad \textit{and}$$
$$L_2 = c \cdot L_b \cup L_a \cdot d \cup L_{even} \cdot d \cup c \cdot L_{even} \cdot d \cup \{a, b\}^*.$$

Then $L_1^{\circledast} \cap L_2^{\circledast} \cap c \cdot L_{even} \cdot d = \{cwwd \mid w \in \{a, b\}^*\}$.

4 Closure properties

Let $\mathcal{L} \subseteq 2^{\Sigma^*}$ be a class of languages. Denote co-$\mathcal{L} = \{\overline{L} \mid L \in \mathcal{L}\}$.

Theorem 8 *A class of languages $\mathcal{L} \subseteq 2^{\Sigma^*}$ is closed under dual concatenation (dual star) if and only if co-$\mathcal{L}$ is closed under concatenation (star, resp.).*

PROOF. Let $\mathcal{L}$ be closed under concatenation, and consider an arbitrary pair of languages $L_1, L_2 \in$ co-$\mathcal{L}$. The languages $\overline{L_1}$ and $\overline{L_2}$ are in $\mathcal{L}$ according to the definition of co-$\mathcal{L}$, and then, by the closure of $\mathcal{L}$ under concatenation, $\overline{L_1} \cdot \overline{L_2}$ is in $\mathcal{L}$. Hence, its complement, $\overline{\overline{L_1} \cdot \overline{L_2}} = L_1 \odot L_2$, is in co-$\mathcal{L}$. The case of dual star is proved in the same way. □

Corollary 2 *Every family of languages closed under complement is (i) either closed or not closed under both concatenation and dual concatenation; (ii) either closed or not closed under both star and dual star.*

Theorem 9 *Regular, deterministic context-sensitive, context-sensitive, recursive, and the languages generated by Boolean grammars [12] are closed under both dual concatenation and dual star. Deterministic context-free and linear conjunctive languages [11] are closed under neither dual concatenation nor dual star.*

PROOF. All these families are known to be closed under complement. The families from the first group are closed under both concatenation and star, while the families from the second group are closed under neither. The results follow from Corollary 2. □

Lemma 1 *Consider the languages L_a, L_b, L_{even} from Example 1. The language $L_a \odot (L_b \cup L_{even})$ is not a finite intersection of context-free languages ($\cap CF$). Similarly, the languages $L_1^{\circledast}$ and $L_2^{\circledast}$ from Example 2 are not in $\cap CF$.*

PROOF. Suppose $L_a \odot (L_b \cup L_{even})$ is in $\cap CF$. By the symmetry, so is $L_b \odot (L_a \cup L_{even})$. Hence, their intersection is also in $\cap CF$. However, this intersection equals $\{ww \mid w \in \{a, b\}^*\}$, which is known not to be in $\cap CF$ [17], a contradiction.

If $L_1^{\circledast}$ from Example 2 (and, by the symmetry, $L_2^{\circledast}$) can be represented by such an intersection, then $L_1^{\circledast} \cap L_2^{\circledast} \cap c \cdot L_{even} \cdot d = \{cwwd \mid w \in \{a, b\}^*\}$ is in $\cap CF$, which similarly yields a contradiction. □

Since L_a, $L_b \cup L_{even}$ and L_1 are linear context-free, Lemma 1, partially with the help of Theorem 8, leads to the following nonclosure results:

Theorem 10 *Context-free and linear context-free languages are not closed under dual concatenation and dual star. Co-context-free and co-linear-context-free languages are not closed under concatenation and star.*

Theorem 11 *Finite languages are closed under both dual concatenation and dual star.*

PROOF. The case of $\odot$. Let L_1 and L_2 be arbitrary finite languages. Let m_i be the length of the longest string in L_i (take $m_i = 0$ if $L_i = \varnothing$) and let us prove that $L_1 \odot L_2$ does not contain strings of more than $m_1 + m_2 + 1$ symbols long.

Suppose there is $w \in L_1 \odot L_2$, such that $|w| \geqslant m_1 + m_2 + 2$. By the definition of dual concatenation, for every factorization $w = uv$ it should hold that $u \in L_1$ or $v \in L_2$. Consider the factorization $w = uv$, such that $|u| = m_1 + 1$ and thus $|v| \geqslant m_2 + 1$. By the choice of m_i, $u \notin L_1$ and $v \notin L_2$, which is a contradiction.

So, $L_1 \odot L_2 \subseteq \Sigma^{\leqslant m_1+m_2+1}$ and is therefore finite.

The case of $\circledast$. By Theorem 7, $L^{\circledast}$ is a subset of L, and hence is finite. $\square$

	$\sim$	$\cup$	$\cap$	$\cdot$	$\odot$	$*$	$\circledast$
Finite	−	+	+	+	+	−	+
co-Finite	−	+	+	+	+	+	−
Regular	+	+		+		+	
LinCF	−	+	−	−	−	−	−
co-LinCF	−	−	+	−	−	−	−
DetCF	+	−		−		−	
CF	−	+	−	+	−	+	−
co-CF	−	−	+	−	+	−	+
Lin.Conj. [8,11]	+	+		−		−	
Conjunctive [8]	?	+	+	+	?	+	?
co-Conjunctive	?	+	+	?	+	?	+
Boolean [12]	+	+		+		+	
DetCS	+	+		+		+	
CS	+	+		+		+	
Recursive	+	+		+		+	
RE	−	+	+	+	+	+	+
co-RE	−	+	+	+	+	+	+

Table 1
Closure properties of common families of languages

Finally, the following theorem can easily be proved by an explicit construction of a Turing machine.

Theorem 12 *Recursively enumerable and co-recursively enumerable languages* *are closed under both dual concatenation and dual star.*

The results of this section are put together in Table 1.

5 Dual concatenation in regular expressions

A fundamental theorem due to Kleene states that a set is recognized by a finite automaton if and only if it can be represented as a *regular expression*, which is a formula over the operations $\cup$, $\cdot$ and $*$ and the constants a, ε and $\varnothing$. More expressive formalisms, *semi-extended regular expressions* with the operations $\{\cup, \cap, \cdot, *\}$, and *extended regular expressions* with $\cup$, $\cap$, $\sim$, $\cdot$ and $*$ [4,18], have been subsequently studied.

Consider the following related formalism:

Definition 8 (Extended dual regular expression) *Let Σ be an alphabet. (i) $\varnothing$, $\overline{\varnothing}$, ε, $\overline{\varepsilon}$, a and $\overline{a}$ (for every $a \in \Sigma$) are extended dual regular expressions; (ii) If α is an extended dual regular expression, then so are α^* and $\alpha^{\circledast}$; (iii) If α and β are extended dual regular expressions, then so are $(\alpha \cdot \beta)$, $(\alpha \odot \beta)$, $(\alpha \mid \beta)$ and $(\alpha \& \beta)$.*

A language $L(\alpha)$ is associated with every extended dual regular expression α as follows: $L(\varepsilon) = \{\varepsilon\}$, $L(\overline{\varepsilon}) = \Sigma^+$, $L(a) = \{a\}$, $L(\overline{a}) = \Sigma^* \setminus \{a\}$, $L(\alpha^*) = L(\alpha)^*$, $L(\alpha^{\circledast}) = L(\alpha)^{\circledast}$, $L(\alpha \cdot \beta) = L(\alpha) \cdot L(\beta)$, $L(\alpha \odot \beta) = L(\alpha) \odot L(\beta)$, $L(\alpha \mid \beta) = L(\alpha) \cup L(\beta)$, $L(\alpha \& \beta) = L(\alpha) \cap L(\beta)$.

As compared with extended regular expressions [18], extended dual regular expressions do not have negation, but have dual concatenation and dual star instead. It turns out that the lack of negation does not increase the descriptional complexity, as any given extended dual regular expression can be negated by simply changing all operations to their duals:

Definition 9 The dual $d(\alpha)$ *of an extended dual regular expression α is defined as $d(\varnothing) = \overline{\varnothing}$, $d(\overline{\varnothing}) = \varnothing$, $d(\varepsilon) = \overline{\varepsilon}$, $d(\overline{\varepsilon}) = \varepsilon$, $d(a) = \overline{a}$, $d(\overline{a}) = a$, $d(\alpha^*) = d(\alpha)^{\circledast}$, $d(\alpha^{\circledast}) = d(\alpha)^*$, $d(\alpha \cdot \beta) = d(\alpha) \odot d(\beta)$, $d(\alpha \odot \beta) = d(\alpha) \cdot d(\beta)$, $d(\alpha \mid \beta) = d(\alpha) \& d(\beta)$, $d(\alpha \& \beta) = d(\alpha) \mid d(\beta)$.*

Lemma 2 *For every extended dual regular expression α, $d(d(\alpha)) \equiv \alpha$ and $L(d(\alpha)) = \overline{L(\alpha)}$.*

So the succinctness of description of languages by extended regular expressions and by extended dual regular expressions is the same. Along with this, the latter inherit many other noteworthy properties of the former, such as the nonelementary complexity of the emptiness problem [4], and nonelementary succinctness tradeoff with the standard regular expressions [4]. It is important to emphasize that in our case this is being achieved without using the complement: the operations $\cdot$, $\odot$, $*$, $\circledast$, $\cup$ and $\cap$ are all monotone.

Let us now consider some restricted classes of regular expressions with dual concatenation: *semi-extended dual regular expressions* forbid the operations of $\cdot$ and $*$ and the constants $\varnothing$, ε and a (retaining $\odot$, $\circledast$, $\cup$, $\cap$ and the constants $\overline{\varnothing}$, $\overline{\varepsilon}$, $\overline{a}$), while *dual regular expressions* also exclude union (thus featuring $\odot$, $\circledast$, $\cap$ and $\overline{\varnothing}$, $\overline{\varepsilon}$, $\overline{a}$).

Proposition 2 *The dual of every regular expression is a dual regular expression. The dual of every semi-extended regular expression is a semi-extended dual regular expression.*

It is easy to see that dual regular expressions have the same expressive power as the standard regular expressions: for every regular L, $\overline{L}$ is regular and hence is generated by some regular expression α; consequently, the dual regular expression $d(\alpha)$ generates L. With respect to the semiring $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+ \rangle$, $\odot$, $\circledast$ and $\cap$ are known as the *rational operations* in this semiring; hence the following statement:

Proposition 3 *The class of languages rational in the semiring* $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+ \rangle$ *is exactly the class of regular languages.*

Thus dual regular expressions, semi-extended dual regular expressions and extended dual regular expressions form three new equivalent representations of regular sets. These representations vary in descriptional complexity:

Example 3 *It is known that the language* $L_n = (0 \,|\, 1)^* \cdot 0 \cdot (0 \,|\, 1)^n \cdot 0 \cdot (0 \,|\, 1)^*$ *is specified by a regular expression of size* $O(n)$*, while* $\overline{L_n}$ *requires a regular expression of size at least* $C \cdot 2^n$ *[4]. On the other hand,* $\overline{L_n}$ *is specified by the dual regular expression* $(\overline{0}\&\overline{1})^{\circledast} \odot \overline{0} \odot (\overline{0}\&\overline{1})^{ⓝ} \odot \overline{0} \odot (\overline{0}\&\overline{1})^{\circledast}$ *of linear size, but* L_n *requires an exponential-size dual regular expression.*

6 Dual concatenation in language equations

Language equations with concatenation and union were studied in the early days of formal language theory as an algebraic semantics for the context-free grammars [3,5]; their extension, additionally equipped with intersection, has been shown to characterize *conjunctive grammars* [8,9] in a similar way.

Complement proved to be a problematic operation for language equations: a system of equations with $\cdot$, $\cup$, $\cap$ and $\sim$ can have no solutions at all or multiple pairwise incomparable solutions, these properties are undecidable, and the expressive power of unique solutions amounts to all recursive languages [10]. Using such equations typically forces one to impose different kinds of constraints [7,12]. In this section, language equations with dual concatenation instead of complement, i.e., with the operations $\cdot$, $\odot$, $\cup$ and $\cap$, will be studied.

6.1 Equations and their solutions

Definition 10 *Let φ_i ($1 \leqslant i \leqslant n$) be expressions that contain constant languages over an alphabet Σ and variables $(X_1, \ldots, X_n)$ that assume values of languages over Σ, connected using language-theoretic operations from a certain fixed set (e.g., $\{\cdot, \odot, \cup, \cap\}$). Then $X_i = \varphi_i(X_1, \ldots, X_n)$ ($X = \varphi(X)$ is vector form) is called a resolved system of language equations. A vector of languages $L = (L_1, \ldots, L_n)$ is its solution if $L_i = \varphi_i(L)$ for all i.*

L is called a solution modulo a language $M \subseteq \Sigma^*$, if $L_i \cap M = \varphi_i(L) \cap M$ for all i (the latter being denoted $L_i = \varphi_i(L) \pmod{M}$). A system is said to have a strongly unique solution, if it has a unique solution modulo every finite language closed under substring.

In light of Theorem 3, it is easy to generalize the well-known properties of language equations over $\{\cdot, \cup\}$ [3,5] for the case of language equations over $\{\cdot, \odot, \cup, \cap\}$:

Theorem 13 *Every system of language equations over $\{\cdot, \odot, \cup, \cap\}$ has a least and a greatest solution given by the limits of the increasing sequence $\{\varphi^i(\varnothing, \ldots, \varnothing)\}_{i=0}^{\infty}$ and of the decreasing sequence $\{\varphi^i(\Sigma^*, \ldots, \Sigma^*)\}_{i=0}^{\infty}$ respectively.*

Thus the decision problems of checking existence and minimality (maximality) of solutions become trivial, while for language equations with explicit complement they are co-RE-complete and Π_2-complete, respectively. The property of having a unique solution remains nontrivial (consider a one-variable resolved equation $X = X$). Let us establish a necessary and sufficient condition.

Lemma 3 (Extension of a solution modulo a language) *Let $X = \varphi(X)$ be a system of language equations over $\{\cdot, \odot, \cup, \cap\}$. If L_M is a solution modulo a finite language M closed under substring, then there exists a solution L of the system, such that $L = L_M \pmod{M}$.*

PROOF. [A sketch] Consider the sequence $\{\varphi^k(L_M)\}_{k=0}^{\infty}$. It can be proved to be increasing, and every term of the sequence equals L_M modulo M, since $\varphi(L_M) = L_M \pmod{M}$. Let $L = \sup_{k \geqslant 0} \varphi^k(L_M)$. L can be proved to be a solution of the system (since φ is $\cup$-continuous), and it equals L_M modulo M. $\square$

Theorem 14 (Criterion of solution uniqueness for monotone systems) *A system over $\{\cdot, \odot, \cup, \cap\}$ has a unique solution if and only if it has a strongly unique* *solution.*

PROOF. In one direction, the proof is trivial: if a system has a strongly unique solution, then it has a unique solution [10]. To prove the converse, let a system have a unique solution and let L'_M, L''_M be solutions modulo some finite M closed under substring. Then, by Lemma 3, there exist solutions L', L'' of the system, such that $L' = L'_M \pmod{M}$ and $L'' = L''_M \pmod{M}$. Since the solution of the system is unique, $L' = L''$, and hence $L'_M = L''_M \pmod{M}$. $\square$

Theorem 14 expresses the uniqueness of a solution by a first-order formula with a single universal quantifier; this shows that the problem is co-recursively-enumerable. On the other hand, the same problem for language equations with union and linear concatenation only is already known to be hard for this class, which is proved by reducing the context-free universality problem [14]. Hence the following completeness result:

Theorem 15 *The set of systems over* $\{\cdot, \odot, \cup, \cap\}$ *that have a unique solution is co-RE-complete.*

6.2 Expressive power of unique solutions

For language equations over $\{\cup, \cdot\}$, the classes of languages representable by their least, greatest and unique solutions are the same: that is the class of the context-free languages [1,5]. On the other hand, for language equations with $\{\cup, \cap, \sim, \cdot\}$ these are three different classes: RE, co-RE and recursive languages, respectively [10]. Let us show that our systems of language equations with dual concatenation but without complement inherit this property from the context-free equations:

Theorem 16 *For every system of language equations over* $\{\cdot, \odot, \cup, \cap\}$ *with constants* ε, $\overline{\varepsilon}$, a, $\overline{a}$ *(for all* $a \in \Sigma$*), there exists and can be effectively constructed a system of language equations over the same set of operations, which has a unique solution, such that the first component of this unique solution coincides with the first component of the original system's least (greatest) solution.*

A proof of Theorem 16 is given in the rest of this section. First, let us define the target form that guarantees the uniqueness of a solution, which generalizes the known notion of a *proper system* [1,9]. Afterwards it will be described how to transform a system to this form.

Definition 11 *Consider the following modified operations:*

$$L_1 \cdot_\varepsilon L_2 = (L_1 \cap \Sigma^+) \cdot (L_2 \cap \Sigma^+) \tag{6a}$$

$$L_1 \odot_\varepsilon L_2 = (L_1 \cup \{\varepsilon\}) \odot (L_2 \cup \{\varepsilon\}) \tag{6b}$$

The set of expressions $\varphi(X_1, \ldots, X_n)$ over $\{\cup, \cap, \cdot_\varepsilon, \odot_\varepsilon\}$ admissible in proper systems is defined inductively as follows:

- *a, $\overline{a}$, ε and $\overline{\varepsilon}$ are admissible in proper systems;*
- *$s_1 \cdot_\varepsilon \ldots \cdot_\varepsilon s_m$, where $m \geqslant 2$ and each s_i is a variable X_j or a symbol $a \in \Sigma$, is admissible in proper systems;*
- *$s_1 \odot_\varepsilon \ldots \odot_\varepsilon s_m$, where $m \geqslant 2$ and each s_i is a variable X_j or a negated symbol $\overline{a}$ ($a \in \Sigma$), is admissible in proper systems;*
- *If ψ and ξ are expressions admissible in proper systems, then so are $\psi \cup \xi$ and $\psi \cap \xi$.*

A system $X_i = \varphi_i(X_1, \ldots, X_n)$ ($1 \leqslant i \leqslant n$) is called proper if every φ_i is admissible in proper systems.

Lemma 4 *Let an expression φ over $\{\cup, \cap, \cdot_\varepsilon, \odot_\varepsilon\}$ be admissible in proper systems. Then, for every $k \geqslant 0$, if $L' = L'' \pmod{\Sigma^{\leqslant k-1}}$, then $\varphi(L') = \varphi(L'') \pmod{\Sigma^{\leqslant k}}$.*

PROOF. Induction on the structure of φ.

- The case of constants is trivial.
- Let $w \in (s_1 \cdot_\varepsilon \ldots \cdot_\varepsilon s_m)(L')$ ($|w| \leqslant k$). Then there exists a factorization $w = u_1 \ldots u_m$ ($u_i \neq \varepsilon$), such that $u_i \in s_i(L')$ for all i. This, since $|u_i| < |w| \leqslant k$ and $L' = L'' \pmod{\Sigma^{\leqslant k-1}}$, implies $u_i \in s_i(L'')$. Hence, $w \in (s_1 \cdot_\varepsilon \ldots \cdot_\varepsilon s_m)(L'')$.
- If $w \in (s_1 \odot_\varepsilon \ldots \odot_\varepsilon s_m)(L')$ ($|w| \leqslant k$), then for every factorization $w = u_1 \ldots u_m$ ($u_i \neq \varepsilon$) there exists i, such that $u_i \in s_i(L')$. By the same argument as above, the latter implies $u_i \in s_i(L'')$. Hence, $w \in (s_1 \odot_\varepsilon \ldots \odot_\varepsilon s_m)(L'')$.
- Let $w \in \psi(L') \cup \xi(L')$. Then $w \in \psi(L')$ or $w \in \xi(L')$. By the induction hypothesis twice, $w \in \psi(L'')$ or $w \in \xi(L'')$, which means $w \in \psi(L'') \cup \xi(L'')$.

  The case of intersection is proved in exactly the same way. $\square$

Theorem 17 *Every proper system of language equations over $\{\cup, \cap, \cdot_\varepsilon, \odot_\varepsilon\}$ has a unique solution.*

PROOF. A system $X = \varphi(X)$ has solutions by Theorem 13. Let L', L'' be solutions of the system, and let us show that $L' = L'' \pmod{\Sigma^{\leqslant k}}$ for every $k \geqslant -1$. The basis, $k = -1$, is true. Turning to the induction step, if $L' = L'' \pmod{\Sigma^{\leqslant k-1}}$, then

$$\varphi(L') = \varphi(L'') \pmod{\Sigma^{\leqslant k}} \tag{7}$$

by Lemma 4. Since L' and L'' are both solutions, $L' = \varphi(L')$ and $L'' = \varphi(L'')$, which, together with (7), implies $L' = L''$ $(\text{mod } \Sigma^{\leqslant k})$. Therefore, $L' = L''$. $\square$

Following is the general schedule of transformation of a system over $\{\cup, \cap, \cdot, \odot\}$ to a proper system:

- first, ensure that the right hand side of every equation is a combination of terms of the form $s_1 \cdot \ldots \cdot s_m$ and $s_1 \odot \ldots \odot s_m$ $(m \geqslant 1)$ with union and intersection;
- second, express $\cdot/\odot$ through $\cdot_\varepsilon/\odot_\varepsilon$;
- third, get rid of unit terms, for which $m = 1$.

The first part can be done by a straightforward decomposition of the right hand sides. The second part is done as follows. Fix a vector of languages L^ε, and define with respect to it:

$$\rho(s_1 \cdot \ldots \cdot s_m) = \bigcup_{\substack{1 \leqslant i_1 < \ldots < i_k \leqslant m \\ \forall j \notin \{i_t\}\ \varepsilon \in s_j(L^\varepsilon)}} s_{i_1} \cdot_\varepsilon \ldots \cdot_\varepsilon s_{i_k} \tag{8a}$$

$$\rho(s_1 \odot \ldots \odot s_m) = \bigcap_{\substack{1 \leqslant i_1 < \ldots < i_k \leqslant m \\ \forall j \notin \{i_t\}\ \varepsilon \in s_j(L^\varepsilon)}} s_{i_1} \odot_\varepsilon \ldots \odot_\varepsilon s_{i_k} \tag{8b}$$

Lemma 5 *Let ρ be defined with respect to L^ε. Let $L = L^\varepsilon$ $(\text{mod } \{\varepsilon\})$. Let φ be of the form $s_1 \cdot \ldots \cdot s_m$ or $s_1 \odot \ldots \odot s_m$ ($m \geqslant 0$, $s_i \in \Sigma \cup X$). Then $\varphi(L) = \rho(\varphi)(L)$.*

The case of concatenation generally repeats the well-known case of removing epsilon rules from context-free grammars (Bar-Hillel, Perles and Shamir [2]). The case of dual concatenation is formally dual to the first case, since $d(\rho(\varphi)) \equiv \rho(d(\varphi))$.

Let us extend the notation ρ (8) to handle unions and intersections of concatenations $s_1 \cdot \ldots \cdot s_m$ and dual concatenations $s_1 \odot \ldots \odot s_m$: define $\rho(\varphi \cup \psi) = \rho(\varphi) \cup \rho(\psi)$ and $\rho(\varphi \cap \psi) = \rho(\varphi) \cap \rho(\psi)$. The statement of Lemma 5 straightforwardly extends to this notation:

Lemma 6 *Under the conditions of Lemma 5, let φ be a finite union of finite intersections of objects of the form $s_1 \cdot \ldots \cdot s_m$ or $s_1 \odot \ldots \odot s_m$. Then $\varphi(L) = \rho(\varphi)(L)$.*

Lemma 7 *Let $X = \varphi(X)$, where φ_i are as in Lemma 6, be a system of language equations, let L^ε be a solution of this system modulo $\{\varepsilon\}$, define ρ with respect to L^ε. Define*

$$\tilde{\varphi}_i = \begin{cases} \rho(\varphi_i) \cup \varepsilon, \text{ if } \varepsilon \in L_i^\varepsilon \\ \rho(\varphi_i) \cap \overline{\varepsilon}, \text{ if } \varepsilon \notin L_i^\varepsilon \end{cases} \tag{9}$$

Then a vector of languages L is a solution of $X = \tilde{\varphi}(X)$ if and only if L is a solution of $X = \varphi(X)$ and $L = L^\varepsilon$ $(\text{mod } \{\varepsilon\})$.

PROOF. $\textcircled{\Rightarrow}$ Let

$$L_i = \tilde{\varphi}_i(L) \tag{10}$$

for all i. By (9), $\varepsilon \in \tilde{\varphi}_i(L)$ if and only if $\varepsilon \in L_i^\varepsilon$; by (10), this implies that $\varepsilon \in L_i$ if and only if $\varepsilon \in L_i^\varepsilon$ (for all i). Therefore, $L = L^\varepsilon \pmod{\{\varepsilon\}}$.

The latter fact allows to apply Lemma 6 to each φ_i and to L, giving

$$\rho(\varphi_i)(L) = \varphi_i(L) \tag{11}$$

According to (9),

$$\tilde{\varphi}_i(L) = \rho(\varphi_i)(L) \pmod{\Sigma^+} \tag{12}$$

Putting together (10), (12) and (11), we obtain

$$L_i = \varphi_i(L) \pmod{\Sigma^+} \tag{13}$$

On the other hand, since L^ε is a solution of $X = \varphi(X)$ modulo $\{\varepsilon\}$ by assumption, and $L = L^\varepsilon \pmod{\{\varepsilon\}}$ as proved above, we infer that $L_i = \varphi_i(L) \pmod{\{\varepsilon\}}$, which, together with (13), proves that $L = \varphi(L)$.

$\textcircled{\Leftarrow}$ Let $L = L^\varepsilon \pmod{\{\varepsilon\}}$ and $L_i = \varphi_i(L)$ for all i. By Lemma 6, $\varphi_i(L) = \rho(\varphi_i)(L)$. Together, this yields

$$L_i = \rho(\varphi_i)(L) \tag{14}$$

If $\varepsilon \in L_i^\varepsilon$, then $L_i = L_i \cup \{\varepsilon\}$. From (14) we obtain $L_i \cup \{\varepsilon\} = \rho(\varphi_i)(L) \cup \{\varepsilon\} = \tilde{\varphi}_i(L)$, and hence $L_i = \tilde{\varphi}_i(L)$.

Similarly, if $\varepsilon \notin L_i^\varepsilon$, then $L_i = L_i \cap \Sigma^+$, while (14) implies $L_i = L_i \cap \Sigma^+ = \rho(\varphi_i)(L) \cap \Sigma^+ = \tilde{\varphi}_i(L)$, and again $L_i = \tilde{\varphi}_i(L)$, which completes the proof. $\square$

For the final, third step of the transformation, the next lemma gives a way to eliminate individual unit terms, which is necessary to obtain a proper system.

Lemma 8 *Let $X = \varphi(X)$ be a system over $\{\cdot_\varepsilon, \odot_\varepsilon, \cup, \cap\}$. Let the equation for X_i be of the form $X_i = (X_i \cap \psi) \cup \xi$. Then, if it is replaced with $X_i = \xi$ ($X_i = \psi \cup \xi$, resp.), the resulting system has the same least (greatest, resp.) solution.*

PROOF. The case of a least solution. Denote the new system as $X = \tilde{\varphi}(X)$. It is claimed that $\tilde{\varphi}^k(\varnothing, \ldots, \varnothing) = \varphi^k(\varnothing, \ldots, \varnothing)$ for all $k \geqslant 0$. We proceed by induction on k. The basis holds; for the induction step, given $L = \tilde{\varphi}^k(\varnothing, \ldots, \varnothing) = \varphi^k(\varnothing, \ldots, \varnothing)$, it suffices to show that $\xi(L) = ((X_i \cap \psi) \cup \xi)(L)$.

The inclusion $\xi(L) \subseteq ((X_i \cap \psi) \cup \xi)(L)$ is evident. Conversely, if $w \in ((X_i \cap \psi) \cup \xi)(L)$, then $w \in (X_i \cap \psi)(L) \subseteq L_i$ or $w \in \xi(L)$. While the latter case is clear, in the former case, since the sequence $\{\tilde{\varphi}^\ell(\varnothing, \ldots, \varnothing)\}$ is increasing, L_i (which equals the i-th component of $\tilde{\varphi}^k(\varnothing, \ldots, \varnothing)$) is a subset of $\xi(L)$ (the i-th component of $\tilde{\varphi}^k(\varnothing, \ldots, \varnothing)$), and again $w \in \xi(L)$.

The case of a greatest solution. Let $X = \tilde{\varphi}(X)$ be the new system, and let us prove that $\tilde{\varphi}^k(\Sigma^*, \ldots, \Sigma^*) = \varphi^k(\Sigma^*, \ldots, \Sigma^*)$ for all $k \geqslant 0$. Induction on k. Basis: true. Induction step: given $L = \tilde{\varphi}^k(\Sigma^*, \ldots, \Sigma^*) = \varphi^k(\Sigma^*, \ldots, \Sigma^*)$, we need to show that $(\psi \cup \xi)(L) = ((X_i \cap \psi) \cup \xi)(L)$.

As in the previous case, $(\psi \cup \xi)(L) \supseteq ((X_i \cap \psi) \cup \xi)(L)$ is obvious. Conversely, if $w \notin ((X_i \cap \psi) \cup \xi)(L)$, then $w \notin (X_i \cap \psi)(L)$ (meaning $w \notin L_i$ or $w \notin \psi(L)$) and $w \notin \xi(L)$. There are two cases to consider: (i) $w \notin \psi(L)$ and $w \notin \xi(L)$: then, as required, $w \notin (\psi \cup \xi)(L)$; (ii) $w \notin L_i$ and $w \notin \xi(L)$: L_i a superset of $(\psi \cup \xi)(L)$, since the sequence $\{\tilde{\varphi}^\ell(\Sigma^*, \ldots, \Sigma^*)\}$ is decreasing, and therefore $w \notin (\psi \cup \xi)(L)$. □

Now the main theorem can be proved.

PROOF. [Proof of Theorem 16] First, decompose the right hand sides of the given system to match Lemma 6. Let $X = \varphi(X)$ be this decomposed system over $\{\cdot, \odot, \cup, \cap\}$. Let L^ε be the least (greatest) solution of this system modulo $\{\varepsilon\}$. Define ρ with respect to L^ε and use Lemma 7 to obtain a system over $\{\cdot_\varepsilon, \odot_\varepsilon, \cup, \cap\}$ with the same least (greatest) solution. Apply Lemma 8 until all unit terms are eliminated. The resulting system is proper and has a unique solution by Theorem 17. Finally, express $\cdot_\varepsilon$, $\odot_\varepsilon$ using (6) to obtain a system over the original set of operations. □

The unique solution constructed in Theorem 16 is, according to Theorem 14, strongly unique. Therefore, least, greatest, unique and strongly unique solutions of language equations over $\{\cdot, \odot, \cup, \cap\}$ specify a common class of languages. In the following it will be shown that this is the class defined by strongly unique solutions of language equations over $\{\cdot, \cup, \cap, \sim\}$.

6.3 Elimination of negation

Definition 12 (cf. Def. 9) *Let Σ be an alphabet, let $\varphi(X_1, \ldots, X_n)$ be an expression over Σ, which uses the operations $\cup, \cap, \cdot, \odot$. The dual of φ, denoted $d(\varphi)$, is defined inductively on the structure of φ as follows: $d(\varepsilon) = \overline{\varepsilon}$, $d(\overline{\varepsilon}) = \varepsilon$,*

$d(a) = \overline{a}$ and $d(\overline{a}) = a$ for every $a \in \Sigma$, $d(X_i) = X_i$, $d(\psi \cdot \xi) = d(\psi) \odot d(\xi)$, $d(\psi \odot \xi) = d(\psi) \cdot d(\xi)$, $d(\psi \cup \xi) = d(\psi) \cap d(\xi)$, $d(\psi \cap \xi) = d(\psi) \cup d(\xi)$.

Lemma 9 (cf. Lemma 2) *For every formula φ, $d(d(\varphi)) = \varphi$, and for every vector of languages $(L_1, \dots, L_n)$, $d(\varphi)(L_1, \dots, L_n) = \overline{\varphi(\overline{L_1}, \dots, \overline{L_n})}$.*

Corollary 3 *$(L_1, \dots, L_n)$ is a solution of $X_i = \varphi_i(X)$ ($i = 1 \dots n$) if and only if $(\overline{L_1}, \dots, \overline{L_n})$ is a solution of $X_i = d(\varphi_i)(X)$ ($i = 1 \dots n$)*

Corollary 4 *If a system $X_i = \varphi_i(X)$ ($i = 1 \dots n$) has a unique (a least, a greatest) solution $(L_1, \dots, L_n)$, then the system $X_i = d(\varphi_i)(X)$ ($i = 1 \dots n$) has the unique (the greatest, the least, resp.) solution $(\overline{L_1}, \dots, \overline{L_n})$.*

Example 4 *Consider the language equation $X = a \cdot X \cdot b \,\cup\, X \cdot X \,\cup\, \varepsilon$. Its least solution is the Dyck language, while its greatest solution is Σ^*. Therefore, the dual equation $X = \overline{a} \odot X \odot \overline{b} \,\cap\, X \odot X \,\cap\, \Sigma^+$ has $\varnothing$ as the least solution and the complement of the Dyck language as the greatest solution.*

Example 5 *Consider the following three one-variable resolved language equations:*

$$X = a \cdot \overline{\overline{\overline{X}^{2}}^{2}}^{2} \qquad X = a \cdot \overline{\overline{\overline{\left(a \cdot \overline{\overline{\overline{X}^{2}}^{2}}^{2}\right)}^{2}}^{2}}^{2} \qquad X = a \cdot \left(\overline{a} \odot X^{\textcircled{2}^{2^{\textcircled{2}}}}\right)^{2^{\textcircled{2}^{2}}}$$

The first of them has been constructed by Leiss [7], who showed that it has the unique solution $L = \{a^n \mid \exists k \geqslant 0,$ such that $2^{3k} \leqslant n < 2^{3k+2}\}$. The second one is a substitution of the Leiss equation into itself: if the first equation is $X = \varphi(X)$, the second one is $X = \varphi(\varphi(X))$; hence L is one of its solutions. The third equation is obtained out of the second one by a symbolic rewriting, and it is easy to prove that it has a unique solution, which is nothing but L.

The purpose of the first step in Example 5 is to make the number of negations even. After that they can be merged with the neighbouring concatenations to form dual concatenations. This idea can be used in a more general context as follows:

Lemma 10 *For every system of language equations over $\{\cdot, \cup, \cap, \sim\}$, which has a strongly unique solution with the first component L, there exists and can be effectively constructed a system of language equations over $\{\cdot, \odot, \cup, \cap\}$, which has a unique solution with the same first component.*

PROOF. It is known [12] that every system of language equations over $\{\cdot, \cup, \cap, \sim\}$ that has a strongly unique solution can be effectively transformed to a form akin to Chomsky normal form for the context-free grammars, in which every equation $X_t = \varphi_t$ has $\varphi_t = \xi_{t1} \cup \dots \cup \xi_{tl_t}$, where each ξ_{tj} is either $a \in \Sigma$, or of the form

	$\cup, \cap, \cdot, \sim$	$\cup, \cap, \cdot, \odot$
Decision problems		
Existence of solution	co-RE-complete [10]	trivial (Th. 13)
Uniqueness of solution	Π_2-complete [10]	co-RE-complete (Th. 15)
Strong uniqueness of solution	co-RE-complete [12]	co-RE-complete (Th. 15)
Existince of minimal solution	Π_2-complete [10]	trivial (Th. 13)
Existence of maximal solution	Π_2-complete [10]	trivial (Th. 13)
Expressive power		
Unique solutions	Recursive [10]	Boolean (Th. 18)
Least solutions	RE [10]	Boolean (Th. 18)
Greatest solutions	co-RE [10]	Boolean (Th. 18)
Strongly unique solutions	Boolean [12]	Boolean (Th. 18)

Table 2
Negation vs. dual concatenation in language equations

$\bigcap_{i=1}^{p} Y_i \cdot Z_i \cap \bigcap_{i=1}^{q} \overline{U_i \cdot V_i} \cap \overline{\varepsilon}$. The intersections with $\overline{\varepsilon}$ effectively forbid ε in the solution of the system; hence, the latter expression can be equivalently rewritten as $\bigcap_{i=1}^{p} Y_i \cdot_{\varepsilon} Z_i \cap \bigcap_{i=1}^{q} \overline{U_i \cdot_{\varepsilon} V_i} \cap \overline{\varepsilon}$. This results in a proper system as in Definition 11.

Let $(X_1, \ldots, X_n)$ be the vector of variables of the original system. Consider the vector of variables $(X_1, \ldots, X_n, X'_1, \ldots, X'_n)$ and let us construct a new system over $\{\cdot_{\varepsilon}, \odot_{\varepsilon}, \cup, \cap\}$ that will have the unique solution $L' = (L_1, \ldots, L_n, \overline{L_1}, \ldots, \overline{L_n})$. For that purpose, define ξ'_{tj} as either a or $\bigcap_{i=1}^{p} Y_i \cdot_{\varepsilon} Z_i \cap \bigcap_{i=1}^{q} U'_i \odot_{\varepsilon} V'_i \cap \overline{\varepsilon}$. It is easy to see that $\xi_{tj}(L_1, \ldots, L_n) = \xi'_{tj}(L_1, \ldots, L_n, \overline{L_1}, \ldots, \overline{L_n})$. Define $\varphi'_t = \xi'_{t1} \cup \ldots \cup \xi'_{tl_t}$. Obviously, $\varphi'_t(L_1, \ldots, L_n, \overline{L_1}, \ldots, \overline{L_n}) = \varphi_t(L_1, \ldots, L_n) = L_t$.

By Lemma 9, $d(\varphi'_t)(\overline{L_1}, \ldots, \overline{L_n}, L_1, \ldots, L_n) = \overline{\varphi'_t(L_1, \ldots, L_n, \overline{L_1}, \ldots, \overline{L_n})} = \overline{L_t}$. Let us swap X_i and X'_i in $d(\varphi'_t)$, thus obtaining the system $\{X_t = \varphi'_t(X_1, \ldots, X_n, X'_1, \ldots, X'_n),\ X'_t = d(\varphi'_t)(X'_1, \ldots, X'_n, X_1, \ldots, X_n)\}$. L' is its solution, and it is unique, because the system is proper. □

Theorem 18 *The set of languages specified by unique (least, greatest) solutions of system of language equations over* $\{\cdot, \odot, \cup, \cap\}$ *coincides with the set of languages specified by strongly unique solutions of systems over* $\{\cdot, \cup, \cap, \sim\}$.

Strongly unique solutions of systems over $\{\cdot, \cup, \cap, \sim\}$ were originally used to define *Boolean grammars* [12], which are context-free grammars with added conjunction and negation. The same class of languages has now been obtained using only monotone operations. The differences between these two representations are summarized in Table 2. Although negation is intuitively more clear than dual concatenation, the formal properties of language equations with dual concatenation are definitely more attractive.

6.4 Restricted cases

Let us now determine the expressive power of restricted types of language equations, which may contain dual concatenation but not concatenation. First, using Corollary 4, a statement akin to Proposition 2 can be obtained.

Proposition 4 *If unique (least, greatest) solutions of language equations over some fixed set of operations define a language family $\mathcal{L}$, then unique (greatest, least, resp.) solutions of language equations over the set of duals of these operations specify co-$\mathcal{L}$.*

Theorem 19 *The languages specified by least, unique and greatest solutions of language equations with $\cap$ and* LIN$\odot$ *are the* ***co-linear-context-free*** *languages; language equations with $\cap$ and $\odot$ specify* ***co-context-free*** *languages; language equations with $\cup$, $\cap$ and* LIN$\odot$ *specify* ***linear conjunctive*** *languages [8,11]; language equations with $\cup$, $\cap$ and $\odot$ specify* ***co-conjunctive*** *languages [8].*

PROOF. It is well-known that least and unique solutions of language equations with $\cup$ and $\cdot$ specify context-free languages [5], and it can be proved that the greatest solutions of these language equations also specify only the context-free lan-

Operations	Strongly unique	Unique	Least	Greatest
$\cup$, LIN$\cdot$	LinCF [12]	LinCF [3,5]	LinCF [3,5]	LinCF
$\cap$, LIN$\odot$	co-LinCF	co-LinCF	co-LinCF	co-LinCF
$\cup, \cdot$	CF [12]	CF [3,5]	CF [3,5]	CF
$\cap, \odot$	co-CF	co-CF	co-CF	co-CF
$\cup, \cap$, LIN$\cdot$	Lin.Conj. [12]	Lin.Conj. [9]	Lin.Conj. [9]	Lin.Conj.
$\cup, \cap$, LIN$\odot$	Lin.Conj.	Lin.Conj.	Lin.Conj.	Lin.Conj.
$\cup, \cap, \cdot$	Conj. [12]	Conj. [9]	Conj. [9]	Conj.
$\cup, \cap, \odot$	co-Conj.	co-Conj.	co-Conj.	co-Conj.
$\cup, \cap, \cdot, \odot$	Bool.	Bool.	Bool.	Bool.
$\cup, \cap, \sim$, LIN$\cdot$	Lin.Conj. [12]	Rec. [10]	RE [10]	co-RE [10]
$\cup, \cap, \sim, \cdot$	Bool. [12]	Rec. [10]	RE [10]	co-RE [10]

Table 3
Classes of languages defined by solutions of language equations

guages; hence, by Proposition 4, greatest, unique and least solutions of language equations with $\cap$ and $\odot$ specify the co-context-free languages. The other three cases are proved in the same way; for the linear conjunctive case, the closure of this family under complement is additionally used. $\square$

Corollary 5 *The class of languages algebraic in the semiring* $\langle 2^{\Sigma^*}, \cap, \odot, \Sigma^*, \Sigma^+\rangle$ *is exactly the class of co-context-free languages.*

Some of the known results on the expressive power of language equations are presented in Table 3 together with those established in this paper.

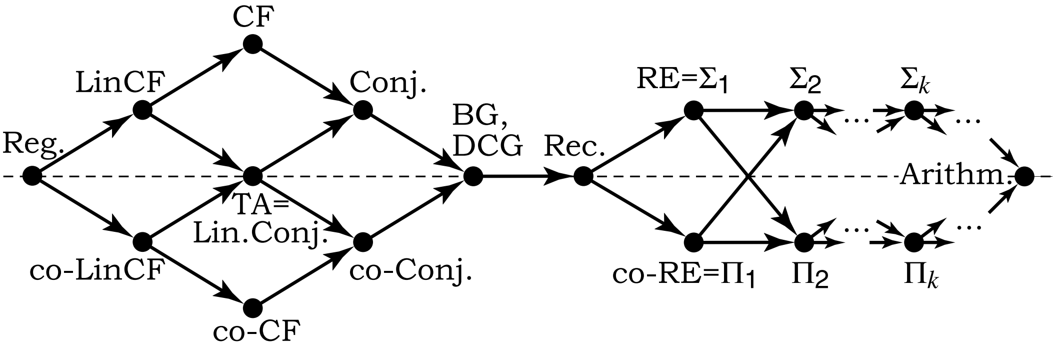

Fig. 2. Hierarchy of language families defined by language equations.

The relation to each other of the families known to be representable by language equations is shown in Figure 2; the arithmetical hierarchy in the right has been characterized using a special partial order on the set of solutions [13]. The symmetry with respect to the dotted line in the figure means complementation; the classes upon the dotted line are closed under complement.

7 Dual concatenation in formal grammars

The principle of *parsing as deduction* [15], brought to a formal perfection in the theory of *parsing schemata* [16], means representing a parsing method as a formal deduction system. For instance, the Cocke–Kasami–Younger algorithm would be formally described using elementary propositions of the form $[A, w]$ ($A \in N$, $w \in \Sigma^+$) and employing axioms like $\vdash [A, a]$ ($A \to a \in P$) and inference rules like $[B, u], [C, v] \vdash [A, uv]$ ($A \to BC \in P$). This approach can also serve as an alternative definition of the context-free grammars. Let us use it as the primary definition of a new family of grammars with explicit dual concatenation.

Definition 13 *A dual concatenation grammar is a quadruple* $G = (\Sigma, N, P, S)$, *in which* Σ *and* N *are finite nonempty disjoint sets of terminal and nonterminal* *systems, $S \in N$ and every production in P is of the form:*

$$A \to s_{11} \cdot \ldots \cdot s_{1k_1} \,\&\, \ldots \,\&\, s_{m1} \cdot \ldots \cdot s_{mk_m} \,\&\, t_{11} \odot \ldots \odot t_{1\ell_1} \,\&\, \ldots \,\&\, t_{n1} \odot \ldots \odot t_{n\ell_n}, \tag{15}$$

where $m + n \geqslant 1$, $k_i, \ell_i \geqslant 0$, $s_{ij} \in \Sigma \cup N$ and $t_{ij} \in \{\overline{a} \mid a \in \Sigma\} \cup N$.

The language generated by a grammar is defined by a formal deduction system:

Definition 14 *Let $\{[\varphi, w] \mid w \in \Sigma^*$ and $\varphi = s_1 \cdot \ldots \cdot s_k$ for some $k \geqslant 0$ and $s_i \in \Sigma \cup N$, **or** $\varphi = s_1 \odot \ldots \odot s_k$ for some $k \geqslant 0$ and $s_i \in \{\overline{a} \,|\, a \in \Sigma\} \cup N\}$ be the set of elementary propositions (items). Define the following axioms:*

A1: $\vdash [\varepsilon, \varepsilon]$,
A2: $\vdash [a, a]$ *(for all $a \in \Sigma$),*
A3: $\vdash [\overline{\varepsilon}, w]$ *(for all $w \in \Sigma^+$),*
A4: $\vdash [\overline{a}, w]$ *(for all $a \in \Sigma$ and $w \in \Sigma^* \setminus \{a\}$).*

Define three types of deduction rules:

C: $[\varphi, u], [\psi, v] \vdash [\varphi \cdot \psi, uv]$ *for all φ, ψ and $u, v \in \Sigma^*$.*

D: $[\varphi_{i_0}, u_0], \ldots, [\varphi_{i_{|w|}}, u_{|w|}] \vdash [\varphi_1 \odot \varphi_2, w]$ *for all φ_1, φ_2, $w \in \Sigma^*$, $i_j \in \{1, 2\}$ and $u_j \in \Sigma^*$ ($0 \leqslant i_j \leqslant |w|$), such that each u_j is a j-symbol prefix of w if $i_j = 1$, or a $(|w| - j)$-symbol suffix of w if $i_j = 2$.*

P: $[s_{11} \cdot \ldots \cdot s_{1k_1}, w], \ldots, [s_{m1} \cdot \ldots \cdot s_{mk_m}, w], [t_{11} \odot \ldots \odot t_{1\ell_1}, w], \ldots, [t_{n1} \odot \ldots \odot t_{n\ell_n}, w] \vdash [A, w]$ *for every rule (15).*

Define $L_G(\varphi) = \{w \mid G \vdash [\varphi, w]\}$ and $L(G) = L_G(S) = \{w \mid G \vdash [S, w]\}$.

Example 6 *Consider the following dual concatenation grammar, which generates the language $\{ww \mid w \in \{a, b\}^*\}$ using the method of Example 1.*

$S \to A \odot D \,\&\, B \odot C$
$A \to a{\cdot}A{\cdot}a \mid a{\cdot}A{\cdot}b \mid b{\cdot}A{\cdot}a \mid b{\cdot}A{\cdot}b \mid a$
$B \to a{\cdot}B{\cdot}a \mid a{\cdot}B{\cdot}b \mid b{\cdot}B{\cdot}a \mid b{\cdot}B{\cdot}b \mid b$
$C \to A \mid E$
$D \to B \mid E$
$E \to a{\cdot}a{\cdot}E \mid a{\cdot}b{\cdot}E \mid b{\cdot}a{\cdot}E \mid b{\cdot}b{\cdot}E \mid \varepsilon$

Let us demonstrate that $G \vdash [S, abab]$:

A2 $\vdash [a, a]$
P $[a, a] \vdash [A, a] \quad (A \to a)$
A2 $\vdash [b, b]$
P $[b, b] \vdash [B, b] \quad (B \to b)$
P $[B, b] \vdash [D, b] \quad (D \to B)$

A1 $\vdash\ [\varepsilon, \varepsilon]$
P $[\varepsilon, \varepsilon]\ \vdash\ [E, \varepsilon] \quad (E \to \varepsilon)$
C $[b, b], [E, \varepsilon]\ \vdash\ [b\cdot E, b]$
C $[a, a], [b\cdot E, b]\ \vdash\ [a\cdot b\cdot E, ab]$
P $[a\cdot b\cdot E, ab]\ \vdash\ [E, ab] \quad (E \to a\cdot b\cdot E)$
C $[b, b], [E, ab]\ \vdash\ [b\cdot E, bab]$
C $[a, a], [b\cdot E, bab]\ \vdash\ [a\cdot b\cdot E, abab]$
P $[a\cdot b\cdot E, abab]\ \vdash\ [E, abab] \quad (E \to a\cdot b\cdot E)$
P $[E, \varepsilon]\ \vdash\ [D, \varepsilon] \quad (D \to E)$
P $[E, ab]\ \vdash\ [D, ab] \quad (D \to E)$
P $[E, abab]\ \vdash\ [D, abab] \quad (D \to E)$
D $[D, abab], [A, a], [D, ab], [D, b], [D, \varepsilon]\ \vdash\ [A \odot D, abab]$
C $[b, b], [A, a]\ \vdash\ [b\cdot A, ba]$
C $[b\cdot A, ba], [b, b]\ \vdash\ [b\cdot A\cdot b, bab]$
P $[b\cdot A\cdot b, bab]\ \vdash\ [A, bab] \quad (A \to bAb)$
P $[A, bab]\ \vdash\ [C, bab] \quad (C \to A)$
C $[a, a], [B, b]\ \vdash\ [a\cdot B, ab]$
C $[a\cdot B, ab], [a, a]\ \vdash\ [a\cdot B\cdot a, aba]$
P $[a\cdot B\cdot a, aba]\ \vdash\ [B, aba] \quad (B \to aBa)$
P $[E, \varepsilon]\ \vdash\ [C, \varepsilon] \quad (C \to E)$
P $[E, ab]\ \vdash\ [C, ab] \quad (C \to E)$
P $[E, abab]\ \vdash\ [C, abab] \quad (C \to E)$
D $[C, abab], [C, bab], [C, ab], [B, aba], [C, \varepsilon]\ \vdash\ [B \odot C, abab]$
P $[A \odot D, abab], [B \odot C, abab]\ \vdash\ [S, abab] \quad (S \to A \odot D \,\&\, B \odot C)$

For dual concatenation grammars there exists a result similar to the well-known characterization of the context-free grammars due to Ginsburg and Rice [5]:

Definition 15 *Let $G = (\Sigma, N, P, S)$ be a dual concatenation grammar. The system of language equations corresponding to G is a system over Σ in variables N, in which the equation for every $A \in N$ is*

$$A = \bigcup_{\substack{\text{There is a rule (15)}\\ \text{for } A \text{ in } P}} \left(\bigcap_{i=1}^{m} (s_{i1} \cdot \ldots \cdot s_{ik_i}) \cap \bigcap_{i=1}^{n} (t_{i1} \odot \ldots \odot t_{i\ell_i}) \right) \tag{16}$$

if there exist any such rules, or $A = aA$ (for some $a \in \Sigma$) if there are none. Let η_A denote the right-hand side of this equation.

Lemma 11 *Let $G = (\Sigma, N, P, S)$ be a dual concatenation grammar, let L be the least solution of the corresponding system of language equations $X = \eta(X)$. Then, for every φ as in Definition 14 and for every $w \in \Sigma^*$, $G \vdash [\varphi, w]$ if and only if $w \in \varphi(L)$.*

PROOF. $\textcircled{\Rightarrow}$ Let $G \vdash [\varphi, w]$. The proof is an induction on the length of the shortest derivation of $[\varphi, w]$. Consider how $[\varphi, w]$ has been derived:

A1, A2, A3, A4. If $\vdash [\varepsilon, \varepsilon]$, then $\varepsilon \in \{\varepsilon\} = \varepsilon(L)$. The proof for the rest of the axioms is also immediate.

C. Let $[\psi, u], [\xi, v] \vdash [\psi \cdot \xi, uv]$ be the last step in the shortest derivation of $[\psi \cdot \xi, uv]$ from the axioms. Then $[\psi, u]$ and $[\xi, v]$ can be derived in fewer steps than $[\psi \cdot \xi, uv]$, and hence, by the induction hypothesis, $u \in \psi(L)$ and $v \in \xi(L)$. This implies $uv \in (\psi \cdot \xi)(L)$

D. Let $[\varphi_{i_0}, u_0], \ldots, [\varphi_{i_{|w|}}, u_{|w|}] \vdash [\varphi_1 \odot \varphi_2, w]$, where $i_j \in \{1, 2\}$ and $u_j \in \Sigma^*$ ($0 \leqslant i_j \leqslant |w|$), such that each u_j is a j-symbol prefix of w if $i_j = 1$, or a $(|w| - j)$-symbol suffix of w if $i_j = 2$. Then, by induction hypothesis, $u_j \in \varphi_{i_j}(L)$ for all j.

Consider an arbitrary factorization $w = xy$, let $j = |x|$. Now x is a j-symbol prefix and y is a $(|w| - j)$-symbol suffix of w. If $i_j = 1$, then we know that $u_j = x \in \varphi_1(L)$; if $i_j = 2$, then $u_j = y \in \varphi_2(L)$. In any case, either $x \in \varphi_1(L)$ or $y \in \varphi_2(L)$. Since this applies for every factorization, $w \in (\varphi_1 \odot \varphi_2)(L)$.

P. Let $[s_{11} \cdot \ldots \cdot s_{1k_1}, w], \ldots, [s_{m1} \cdot \ldots \cdot s_{mk_m}, w], [t_{11} \odot \ldots \odot t_{1\ell_1}, w], \ldots, [t_{n1} \odot \ldots \odot t_{n\ell_n}, w] \vdash [A, w]$ for some rule (15) in P. Then each $[s_{i1} \cdot \ldots \cdot s_{ik_i}, w]$ and each $[t_{i1} \odot \ldots \odot t_{i\ell_i}, w]$ is derivable in fewer steps than $[A, w]$. By the induction hypothesis, $w \in (s_{i1} \cdot \ldots \cdot s_{ik_i})(L)$ for all $1 \leqslant i \leqslant m$ and $w \in (t_{i1} \odot \ldots \odot t_{i\ell_i})(L)$ for all $1 \leqslant i \leqslant n$. Then $w \in (s_{11} \cdot \ldots \cdot s_{1k_1} \& s_{m1} \cdot \ldots \cdot s_{mk_m} \& t_{11} \odot \ldots \odot t_{1\ell_1} \& t_{n1} \odot \ldots \odot t_{n\ell_n})(L) \subseteq \eta_A(L) = A(L)$, where η_A is the right-hand side of (16). The latter equality is due to the fact that L is a solution of $X = \eta(X)$.

$\textcircled{\Leftarrow}$ Denote $L^{(k)} = \eta^k(\varnothing, \ldots, \varnothing)$. If $w \in \varphi(L)$, then, by the convergence of the sequence $\{L^{(k)}\}$ to L, there exists $k \geqslant 0$ such that $w \in \varphi(L^{(k)})$. Inductively on $k \geqslant 0$, let us prove that $G \vdash [\varphi, w]$. The inner induction is on the structure on φ.

- If $\varphi = \varepsilon$, then w must be ε, and the item $[\varepsilon, \varepsilon]$ is derived by **A1**. The proof for the cases $\varphi = a, \overline{\varepsilon}, \overline{a}$ is also immediate.
- If $w \in (\psi \cdot \xi)(L^{(k)})$, there exists a factorization $w = uv$, such that $u \in \psi(L^{(k)})$ and $v \in \xi(L^{(k)})$. By the inner induction hypothesis, $G \vdash [\psi, u]$ and $G \vdash [\xi, v]$. Use the rule **C** to deduce $[\psi \cdot \xi, uv]$.
- If $w \in (\psi \odot \xi)(L^{(k)})$, then for every factorization $w = uv$, $u \in \psi(L^{(k)})$ or $v \in \xi(L^{(k)})$. By the inner induction hypothesis, $G \vdash [\psi, u]$ or $G \vdash [\xi, v]$. This yields a collection of items that is enough to deduce $[\psi \odot \xi, uv]$ using the rule **D**.
- If $w \in A_i(L^{(k)})$, then $w \in L_i^{(k)} = \eta_i(L^{(k-1)})$. Then there exists a rule (15), such that $w \in (s_{i1} \cdot \ldots \cdot s_{ik_i})(L^{(k-1)})$ for all i and $w \in (t_{i1} \odot \ldots \odot t_{i\ell_i})(L^{(k-1)})$ for all i. By the outer induction hypothesis, $G \vdash [s_{i1} \cdot \ldots \cdot s_{ik_i}, w]$ and each $G \vdash [t_{i1} \odot \ldots \odot t_{i\ell_i}, w]$. Hence, by the deduction rule **P**, $G \vdash [A, w]$. $\square$

Theorem 20 *For every dual concatenation grammar $G = (\Sigma, \{A_1, \ldots, A_n\}, P, S)$, the least solution of the corresponding system of equations is $(L_G(A_1), \ldots, L_G(A_n))$.*

This correspondence allows us to use all the results of Section 6 to determine the expressive power of different types of dual concatenation grammars.

Theorem 21 *I. Dual concatenation grammars have the same generative power as Boolean grammars [12].*

II. Consider a subclass of dual concatenation grammars, in which k is the maximal number of rules for a single nonterminal, m is the maximal number of concatenation conjuncts in a rule, n is the maximal number of dual concatenation conjuncts in a rule and ℓ is the maximal number of nonterminals in a conjunct. For different restrictions on k, m, n, ℓ, the classes of languages generated are as follows:

- *context-free, if* $k \geqslant 1$, $m = 1$, $n = 0$, $\ell \geqslant 1$*;*
- *linear context-free, if* $k \geqslant 1$, $m = 1$, $n = 0$, $\ell = 1$*;*
- *conjunctive, if* $k \geqslant 1$, $m \geqslant 1$, $n = 0$, $\ell \geqslant 1$*;*
- *linear conjunctive, if* $k \geqslant 1$, $m \geqslant 1$, $n = 0$, $\ell = 1$*, or, alternatively, if* $k \geqslant 1$, $m = 0$, $n \geqslant 1$, $\ell = 1$*;*
- *co-context-free, if* $k = 1$, $m = 0$, $n \geqslant 1$, $\ell \geqslant 1$*;*
- *co-linear-context-free, if* $k = 1$, $m = 0$, $n \geqslant 1$, $\ell = 1$*;*
- *co-conjunctive, if* $k \geqslant 1$, $m = 0$, $n \geqslant 1$, $\ell \geqslant 1$.

The first part of the proof follows from Theorem 18 and Theorem 20, while the second part is given by Theorems 19 and 20.

Note that the form of the rules (15) in the aforementioned subclasses of dual concatenation grammars that generate (linear) context-free and (linear) conjunctive languages is exactly the same as the form of the rules in the original (linear) context-free and (linear) conjunctive grammars. Hence dual concatenation grammars, despite their differently defined formal semantics, can be viewed as a generalization of these types of grammars. Then it is easy to see that the result of Theorem 20 is not just similar to the characterization of context-free grammars by language equations [3,5], but constitutes its generalization.

8 Conclusion

Joint use of concatenation and complement yields dual concatenation whether we want it or not. This paper attempted to consider this operation as a self-contained notion and to use it in some contexts where both concatenation and complement are naturally used: in regular expressions, in language equations and in formal grammars.

When concatenation and dual concatenation are used *together*, the use of the complement is considerably facilitated. One can reduce the scope of negation to minimal terms, and in many cases get rid of it entirely. The latter was the case with extended dual regular expressions. As for language equations, it turned out that the equations with $\{\cdot, \odot, \cup, \cap\}$ define the same class of languages as a specifically restricted class of equations over $\{\cdot, \cup, \cap, \sim\}$ used in the definition of Boolean grammars [12]. These results also allowed to characterize Boolean grammars by a simple formal deductive system, contributing to the study of this noteworthy language family.

When dual concatenation is used *instead* of the standard concatenation, the resulting constructs tend to specify complements of what could be originally defined. Dual regular expressions form a straightforward example. Among the more elaborate findings are two "positive", negation-free characterizations of the complements of the context-free languages: as the languages algebraic in a certain semiring, and as the languages generated by the proposed *co-context-free grammars*.

Generally, it can be concluded that dual concatenation often allows to avoid the use of logical negation while reasoning about concatenation of languages, or at least to facilitate such reasoning. Hence, wherever concatenation and complement of languages are being used together, it makes sense to recognize the dual concatenation, to denote it explicitly and to take advantage of its properties.

Acknowledgements

I am grateful to Werner Kuich for his inspiring comments on the conference version of this paper and to Galina Jirásková for her helpful remarks on the final manuscript. Special thanks are due to Kai Salomaa for his advices on my Ph. D. thesis, which included this work.

References

[1] J. Autebert, J. Berstel and L. Boasson, "Context-free languages and pushdown automata", *Handbook of formal languages*, Vol. 1, 111–174, Springer, Berlin, 1997.

[2] Y. Bar-Hillel, M. Perles, E. Shamir, "On formal properties of simple phrase-structure grammars", *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14 (1961), 143–177.

[3] N. Chomsky, M. P. Schützenberger, "The algebraic theory of context-free languages", in: Braffort, Hirschberg (Eds.), *Computer Programming and Formal Systems*, North-Holland Publishing Company, Amsterdam, 1963, 118–161.

[4] K. Ellul, J. Shallit, M.-w. Wang, "Regular expressions: new results and open problems", *Pre-proceedings of DCFS 2002* (London, Canada, 21–24 August, 2002),

17–34.

[5] S. Ginsburg, H. G. Rice, "Two families of languages related to ALGOL", *Journal of the ACM*, 9 (1962), 350–371.

[6] W. Kuich, "Semirings and formal power series: their relevance to formal languages and automata", in: Rozenberg, Salomaa (Eds.), *Handbook of Formal Languages*, Vol. 1, Springer-Verlag, 1997, 609–677.

[7] E. L. Leiss, "Unrestricted complementation in language equations over a one-letter alphabet", *Theoretical Computer Science*, 132 (1994), 71–93.

[8] A. Okhotin, "Conjunctive grammars", *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.

[9] A. Okhotin, "Conjunctive grammars and systems of language equations", *Programming and Computer Software*, 28:5 (2002) 243–249.

[10] A. Okhotin, "Decision problems for language equations with Boolean operations", *Automata, Languages and Programming* (Proc. of ICALP 2003, Eindhoven, the Netherlands, June 30–July 4, 2003), LNCS 2719, 239–251; journal version submitted.

[11] A. Okhotin, "On the equivalence of linear conjunctive grammars to trellis automata", *Informatique Théorique et Applications*, 38:1 (2004), 69–88.

[12] A. Okhotin, "Boolean grammars", *Information and Computation*, 194:1 (2004), 19–48.

[13] A. Okhotin, "A characterization of the arithmetical hierarchy by language equations", *Pre-proceedings of DCFS 2004* (London, Ontario, Canada, July 26–28, 2004), 225–237.

[14] A. Okhotin, "Unresolved systems of language equations: expressive power and decision problems", submitted.

[15] S. M. Shieber, Y. Schabes, F. C. N. Pereira, "Principles and implementation of deductive parsing", *Journal of Logic Programming*, 24 (1995), 3–36.

[16] K. Sikkel, *Parsing Schemata*, Springer-Verlag, 1997.

[17] D. Wotschke, "The Boolean closures of deterministic and nondeterministic context-free languages", In: W. Brauer (ed.), *Gesellschaft für Informatik e. V., 3. Jahrestagung 1973*, LNCS 1, Springer-Verlag, Berlin, 1973, 113–121.

[18] S. Yu, "Regular Languages", in: Rozenberg, Salomaa (Eds.), *Handbook of Formal Languages*, Vol. 1, 41–110, Springer-Verlag, Berlin, 1997.