Co-Regular Languages and Unification of Co-Regular Languages

Stoyan Mihov

Linguistic Modelling Laboratory
Bulgarian Academy of Sciences
stoyan@bgcict.acad.bg

draft, comments welcome

Abstract. The main goal of this work is the investigation of some properties of co-regular languages. In many applications it is important to handle infinite sets of terms. We represent two equivalent formalisms for finite defined infinite sets of term substitutions which are leading to co-regular languages. As it will be proved later the co-regular languages are monoidic homomorphical images of regular languages. A natural definition of unification of co-regular languages is introduced. This paper is a compressed version of [Mi93].

1 Monoids

In this section we present a brief introduction in the monoid theory.

Definition 1. A *monoid* $\mathcal{M}$ (semigroup with unit element) is the triple $\langle M, \circ, \mathbf{e}\rangle$ where:

- M is the set of monoid elements;
- $\circ : M \times M \rightarrow M$ is the monoid operation (we will use infix notation);
- $\mathbf{e} \in M$ is the monoid unit element,

and the following conditions hold:

- $\forall a, b, c \in M(a \circ (b \circ c) = (a \circ b) \circ c)$ (associativity),
- $\forall a \in M(a \circ \mathbf{e} = \mathbf{e} \circ a = a)$.

Example 1. The set of finite words over an alphabet Σ with concatenation as monoid operation and the empty word ε as unit element is the free monoid $\Sigma^* = \langle \Sigma^*, *, \varepsilon\rangle$.

Definition 2. Let $\mathcal{M} = \langle M, \circ, \mathbf{e}\rangle$ is a monoid.

1. For $T_1, T_2 \subseteq M$, we define: $T_1 \circ T_2 = \{t_1 \circ t_2 \,|\, t_1 \in T_1, t_2 \in T2\}$.
2. $I \subseteq M$ is an (left) *ideal*, if $M \circ I = M$.
3. We define $Id(T) = M \circ T$, where $T \subseteq M$.

Proposition 3. *For every $T \subseteq M$ holds that $Id(T)$ is an ideal.*

Definition 4. Let $\mathcal{M} = \langle M, \circ, \mathbf{e}\rangle$ is a monoid and $T \subseteq M$, then for every $n \in \mathbb{N}$, we define T^n inductively:

1. $T^0 = \{\mathbf{e}\}$
2. $T^{k+1} = T^k \circ T$

   and $T^+ = \bigcup_{k=1}^{\infty} T^k$, $T^* = \bigcup_{k=0}^{\infty} T^k$. We call T^* the iteration of T.

Definition 5. Let $\mathcal{M}_1 = \langle M_1, \circ, \mathbf{e}_1\rangle$ and $\mathcal{M}_2 = \langle M_2, \bullet, \mathbf{e}_2\rangle$ are monoids. Then the function $\varphi : M_1 \rightarrow M_2$ is a monoid homomorphism when the following conditions hold:

- $\varphi(\mathbf{e}_1) = \mathbf{e}_2$
- $\forall a, b \in M(\varphi(a \circ b) = \varphi(a) \bullet \varphi(b))$

  Let $T \subseteq M_1$, then with $\varphi(T) \subseteq M_2$ we will denote the set $\{\varphi(m) \,|\, m \in T\}$.

Proposition 6. *Let $\varphi : M_1 \to M_2$ is a monoid homomorphism.*

1. *Let $T_\alpha \subseteq M_1$, for every $\alpha \in A \Longrightarrow \varphi(\bigcup_{\alpha\in A} T_\alpha) = \bigcup_{\alpha\in A} \varphi(T_\alpha)$*
2. *Let $T_1, T_2 \subseteq M_1 \Longrightarrow \varphi(T_1 \circ T_2) = \varphi(T_1) \bullet \varphi(T_2)$*
3. *Let $T \in M_1 \Longrightarrow \varphi(T^*) = \varphi(T)^*$*

Proof. The proofs of Points 1, 2 are obtained straight forward. For the proof of Point 3 we have to do an induction and use Points 1, 2. □

Proposition 7. *Let $\mathcal{M} = \langle M, \circ, \mathbf{e}\rangle$ is a monoid, Σ is an alphabet and $\varphi : \Sigma \to M$ is a function. Then the natural extension ϕ of φ over Σ^* define as follows:*

1. *$\phi(\varepsilon) = \mathbf{e}$*
2. *$\phi(\sigma a) = \phi(\sigma) \circ \varphi(a)$, where $\sigma \in \Sigma^*, a \in \Sigma$,*

is the only homomorphism between the monoids Σ^* and $\mathcal{M}$ which is an extension of φ.

2 Terms and the Term Substitution Monoid

This chapter describes our notation for terms and term substitutions.

Let X is a countable set of variables. Let $r : F \to \mathbb{N}$ is the rank function over the set of functional symbols F.

Definition 8. The set of terms $T(X, F)$ we define inductively:

1. $x \in X \Longrightarrow x \in T(X, F)$
2. $f \in F$; $r(f) = k$; $t_1, t_2, \ldots, t_k \in T(X, F) \Longrightarrow f(t_1, t_2, \ldots, t_k) \in T(X, F)$

Definition 9. The set of term substitutions $\mathcal{T}(X, F)$ we define as follows:

$$\mathcal{T}(X, F) = \{\tau : X \to T(X, F) \mid \left|\{x \in X \mid \tau(x) \neq x\}\right| < \infty\}$$

Definition 10. Let $\tau \in \mathcal{T}(X, F)$ and $t \in T(X, F)$, then we extend naturally τ over $T(X, F)$ as follows:

1. $t = x \in X \Longrightarrow \tau(t) = \tau(x)$
2. $t = f(t_1, t_2, \ldots, t_k) \Longrightarrow \tau(t) = f(\tau(t_1), \tau(t_2), \ldots, \tau(t_k))$.

Remark. The composition of substitutions we denote as follows:

$$\text{Let } \tau_1, \tau_2 \in \mathcal{T}(X, F),\ t \in T(X, F) \Longrightarrow (\tau_1 \circ \tau_2)(t) = \tau_1(\tau_2(t)).$$

Proposition 11. $\mathcal{T}(X, F) = \langle \mathcal{T}(X, F), \circ, \mathbf{id}\rangle$ *is a monoid.* (**id** is the identity function over $T(X, F)$.)

Proof.

1. Let $\tau_1, \tau_2 \in \mathcal{T}(X, F)$ then $\tau_1 \circ \tau_2 \in \mathcal{T}(X, F)$ because:

$$\left|\{x \mid \tau_1 \circ \tau_2(x) \neq x\}\right| < \left|\{x \mid \tau_1(x) \neq x\}\right| + \left|\{x \mid \tau_2(x) \neq x\}\right| < \infty$$

2. Obviously $\forall \tau \in \mathcal{T}(X, F)(\tau \circ \mathbf{id} = \mathbf{id} \circ \tau = \tau)$
3. Obviously the operation "$\circ$" viewed as composition of functions over $T(X, F)$ is associative. □

Definition 12. The term t_1 (the substitution τ_1) is more general then the term t_2 (the substitution τ_2 respectively) — $t_1 \leq t_2$ ($\tau_1 \leq \tau_2$) if there exists a substitution $\sigma \in \mathcal{T}(X, F)$ such that: $\sigma(t_1) = t_2$ ($\sigma \circ \tau_1 = \tau_2$).

Proposition 13. $\tau_1 \leq \tau_2 \Longleftrightarrow Id(\{\tau_1\}) \supseteq Id(\{\tau_2\})$.

Definition 14. The depth of the term t — d(t) we define inductively:

1. $t \in X \Longrightarrow d(t) = 0$
2. $t = f(t_1, t_2, \ldots, t_n) \Longrightarrow d(t) = \max\{d(t_i) \mid i = 1, 2, \ldots, n\} + 1$, where when $n = 0$ we assume that $\max \emptyset = 0$.

Definition 15. The depth of the substitution τ — $d(\tau)$ we define as follows:

$$d(\tau) = \max\{d(\tau(x)) \mid x \in X\}\}.$$

Proposition 16. *Let $t \in T(X, F)$ and $\tau, \sigma \in \mathcal{T}(X, F)$. Then the following holds:*

$$d(t) < d(\sigma(t)) < d(\sigma) + d(t)$$
$$d(\tau) < d(\sigma \circ \tau) < d(\sigma) + d(\tau).$$

Proposition 17.

1. *Let $t_1, t_2 \in T(X, F)$ and $t_1 \leq t_2 \Longrightarrow d(t_1) \leq d(t_2)$*
2. *Let $\tau_1, \tau_2 \in \mathcal{T}(X, F)$ and $\tau_1 \leq \tau_2 \Longrightarrow d(\tau_1) \leq d(\tau_2)$.*

Proposition 18. *Let $|X| < \infty$ and $|F| < \infty$. Then the following holds:*

1. $\forall n \in \mathbb{N}(\big|\{t \in T(X, F) \mid d(t) = n\}\big| < \infty)$
2. *Let $t \in T(X, F) \Longrightarrow \big|\{s \in T(X, F) \mid s \leq t\}\big| < \infty$*
3. *Let $\tau \in \mathcal{T}(X, F) \Longrightarrow \big|\{\sigma \in \mathcal{T}(X, F) \mid \sigma \leq \tau\}\big| < \infty$*

Definition 19.

1. The terms $t_1, t_2 \in T(X, F)$ are unifiable, if there exists $\sigma \in \mathcal{T}(X, F)$, such that $\sigma(t_1) = \sigma(t_2)$.
2. The substitutions $\tau_1, \tau_2 \in \mathcal{T}(X, F)$ are unifiable, if there exists $\sigma \in \mathcal{T}(X, F)$, such that $\sigma \circ \tau_1 = \sigma \circ \tau_2$.
3. The sets of substitutions $T_1, T_2 \subseteq \mathcal{T}(X, F)$ are unifiable, if there exists $\sigma \in \mathcal{T}(X, F)$, such that $\sigma \circ T_1 = \sigma \circ T_2$. (With $\tau \circ T$ we denote — $\{\tau\} \circ T$.)

Definition 20.

1. The terms $t_1, t_2 \in T(X, F)$ are weak unifiable, if there exists $\sigma_1, \sigma_1 \in \mathcal{T}(X, F)$, such that $\sigma_1(t_1) = \sigma_2(t_2)$.
2. The substitutions $\tau_1, \tau_2 \in \mathcal{T}(X, F)$ are weak unifiable, if there exists $\sigma_1, \sigma_2 \in \mathcal{T}(X, F)$, such that $\sigma_1 \circ \tau_1 = \sigma_2 \circ \tau_2$.
3. The sets of substitutions $T_1, T_2 \subseteq \mathcal{T}(X, F)$ are weak unifiable, if there exists $\sigma_1, \sigma_2 \in \mathcal{T}(X, F)$, such that $\sigma_1 \circ T_1 = \sigma_2 \circ T_2$.

Remark. The concept of weak unification of terms and term substitutions is described in [Ed85].

Theorem 21. *The problem of term (substitution) unification or weak unification is decidable.*

Proof. In [Ro65] there is an algorithm for term unification. It is easy to modify it for substitution unification and to adapt it for weak unification by variable renaming. □

Proposition 22.

1. *$\tau_1, \tau_2 \in \mathcal{T}(X, F)$ are weak unifiable $\Longleftrightarrow Id(\{\tau_1\}) \cap Id(\{\tau_2\}) \neq \emptyset$*
2. *$T_1, T_2 \subseteq \mathcal{T}(X, F)$ are weak unifiable $\Longleftrightarrow Id(T_1) \cap Id(T_2) \neq \emptyset$*

3 Co-Regular Languages and Monoid Transducers

This chapter introduces the co-regular languages and the monoid transducer languages. As it will be proofed in Theorem 31 both formalisms lead to the same class languages.

Definition 23. Let $\mathcal{M} = \langle M, \circ, \mathbf{e} \rangle$ is a monoid. We define inductively a co-regular language over the monoid $\mathcal{M}$:

1. $\emptyset$ is a co-regular language
2. if $m \in M$, then $\{m\}$ is a coregular language
3. if $L_1, L_2 \subseteq M$ are co-regular languages, then:
   (a) $L_1 \circ L_2$,
   (b) $L_1 \cup L_2$,
   (c) L_1^*
   are co-regular languages.

Remark. In the case when $\mathcal{M}$ is a free monoid — Σ^* the definition above defines the regular languages.

Theorem 24. *Let $T \subseteq \Sigma^*$ is a regular language and $\varphi : \Sigma^* \to M$ is a homomorphism. Then $\varphi(T)$ is a co-regular language.*

Proof. This theorem is easy proofed by induction on the building of the regular language T using Property 6. □

Theorem 25. *Let $L \subseteq M$ is a co-regular language. Then there exist a finite alphabet Σ, homomorphism $\phi : \Sigma^* \to M$ and a regular language $T \subseteq \Sigma^*$, such that $\phi(T) = L$.*

Proof. We will use induction over the definition of L:

1. Let $L = \emptyset$, then for $\Sigma = \emptyset$ and the regular language $T = \emptyset$ we obtain: $\phi(T) = \emptyset = L$.
2. Let $L = \{m\}$, $m \in M$, then for $\Sigma = \{a_m\}$ we define $\varphi : \Sigma \to M$, such that $\varphi(a_m) = m$. Using Proposition 7 we extend φ to homomorphism $\phi : \Sigma^* \to M$. Then for the regular language $T = \{a_m\}$ we obtain:
$$\phi(T) = \phi(\{a_m\}) = \{m\} = L.$$
3. Let $L_1, L_2 \subseteq M$ are co-regular languages. Then from the induction assumption there exist two finite alphabets Σ_1, Σ_2, regular languages $T_1 \subseteq \Sigma_1^*, T_2 \subseteq \Sigma_2^*$ and functions $\varphi_1 : \Sigma_1 \to M, \varphi_2 : \Sigma_2 \to M$ which have been extended to homomorphisms $\phi_1 : \Sigma_1^* \to M, \phi_2 : \Sigma_2^* \to M$ by Proposition 7, such that: $\phi(T_1) = L_1, \phi(T_2) = L_2$.
   (a) if $L = L_1 \cup L_2$ ($L = L_1 \circ L_2$), then we set $\Sigma = \Sigma_1 \cup \Sigma_2$. Because of the compatibility of φ_1 and φ_2 which means that
$$\varphi_1\big|_{\Sigma_1 \cap \Sigma_2} = \varphi_2\big|_{\Sigma_1 \cap \Sigma_2}$$
   we define $\varphi : \Sigma \to M$, such that:
$$\varphi\big|_{\Sigma_1} = \varphi_1, \quad \varphi\big|_{\Sigma_2} = \varphi_2.$$
   Then for the extension $\phi : \Sigma^* \to M$ on φ holds:
$$\phi\big|_{\Sigma_1^*} = \phi_1, \quad \phi\big|_{\Sigma_2^*} = \phi_2.$$
   T_1, T_2 are regular languages over Σ^*, so $T = T_1 \cup T_2$ ($T = T_1 * T_2$) is a regular language over Σ^*. In this way we obtain:
$$\phi(T) = \phi(T_1 \cup T_2) \stackrel{Prop.6}{=} \phi(T_1) \cup \phi(T_2) = L_1 \cup L_2 = L$$
$$\phi(T) = \phi(T_1 * T_2) \stackrel{Prop.6}{=} \phi(T_1) \circ \phi(T_2) = L_1 \circ L_2 = L.$$

(b) If $L = L_1^*$, then for the finite alphabet $\Sigma = \Sigma_1$, the homomorphism $\phi = \phi_1$ and the regular language $T = T_1^*$ holds:

$$\phi(T) = \phi(T_1^*) \overset{Prop.6}{=} \phi(T_1)^* = L_1^* = L.$$

$\square$

Let Σ is a finite alphabet.

Definition 26. Let $\mathcal{M} = \langle M, \circ, \mathbf{e} \rangle$ is a monoid. Monoid transducer over $\mathcal{M}$ is the tuple $\mathcal{T} = \langle S, s_0, S', \mu, \lambda \rangle$, where:

- S is the finite set of transducer positions;
- $s_0 \in S$ is the starting position;
- $S' \subseteq S$ is the set of terminal positions;
- $\mu : S \times \Sigma \to S$ is the transition function;
- $\lambda : S \times \Sigma \to M$ is the output function.

The functions μ and λ are naturally extended over $S \times \Sigma^*$ inductively:

$$\begin{cases} \mu(s, \varepsilon) = s \\ \mu(s, \sigma a) = \mu(\mu(s, \sigma), a) \end{cases} \quad \begin{cases} \lambda(s, \varepsilon) = \mathbf{e} \\ \lambda(s, \sigma a) = \lambda(s, \sigma) \circ \lambda(\mu(s, \sigma), a) \end{cases},$$

where $\sigma \in \Sigma^*, a \in \Sigma$.

Definition 27.

1. The set of terminating paths of the monoid transducer $\mathcal{T}$ is defined as follows:

$$A(\mathcal{T}) = \{\sigma \in \Sigma^* \mid \mu(s_0, \sigma) \in S'\}.$$

2. The language of the monoid transducer $\mathcal{T}$ is the set:

$$L(\mathcal{T}) = \{m \in M \mid \exists \sigma \in A(\mathcal{T}) \, (\lambda(s_0, \sigma) = m)\}.$$

Remark. $A(\mathcal{T})$ is a regular language over Σ^* because $A(\mathcal{T})$ is the language of a Deterministic Finite Automata.,

Remark. In the case $\mathcal{M}$ is a free monoid, we will call the monoid transducer $\mathcal{T}$ a classical transducer.

Remark. In the case $\mathcal{M}$ is $\mathcal{T}(X, F)$, then the definition of the monoid transducer coincide with the definition of the program machine introduced by [Bu78].

Theorem 28. *Let $\mathcal{K} = \langle S, s_0, S', \mu, \lambda \rangle$ is a classical transducer over Σ_0^* and $\varphi : \Sigma_0^* \to M$ is a homomorphism. Then there exists a monoid transducer $\mathcal{T}$ over $\mathcal{M}$, such that $\varphi(L(\mathcal{K})) = L(\mathcal{T})$.*

Proof. We define $\mathcal{T} = \langle S, s_0, S', \mu, \lambda' \rangle$, where $\lambda'(s, a) = \varphi(\lambda(s, a))$ for $s \in S, a \in \Sigma$. With induction it is easy to see that $\varphi(\lambda(s, \sigma)) = \lambda'(s, \sigma)$ for $\sigma \in \Sigma^*$. It is clear that $A(\mathcal{T}) = A(\mathcal{K})$. In this way we obtain $\varphi(L(\mathcal{K})) = L(\mathcal{T})$. $\square$

Theorem 29. *Let $\mathcal{T} = \langle S, s_0, S', \mu, \lambda \rangle$ is a monoid transducer over $\mathcal{M}$. Then there exist a finite alphabet Σ_0, classical transducer $\mathcal{K}$ over Σ_0^* and a homomorphism $\phi : \Sigma_0^* \to M$, such that: $\phi(L(\mathcal{K})) = L(\mathcal{T})$.*

Proof. Let $\mathcal{K} = \langle S, s_0, S', \mu, \lambda' \rangle$ is a classical transducer over Σ_0^*, where:

- $\Sigma_0 = \{a_{\lambda(s,a)} \mid s \in S, a \in \Sigma\}$. Without loose of generality $\Sigma \cap \Sigma_0 = \emptyset$.
- The function λ' is defined as follows: $\lambda'(s, a) = a_{\lambda(s,a)}$

Let we define the function $\varphi : \Sigma_0 \to M$, so that $\varphi(a_{\lambda(s,a)}) = \lambda(s, a)$. Using the Proposition 7 we extend φ to homomorphism $\phi : \Sigma_0^* \to M$. With induction it is easy seen that $\phi(\lambda'(s, \sigma)) = \lambda(s, \sigma)$ for $\sigma \in \Sigma^*$. It is clear that $A(\mathcal{T}) = A(\mathcal{K})$. In this way we obtain $\phi(L(\mathcal{K})) = L(\mathcal{T})$. $\square$

We will use the following variant of the classical Kleene theorem:

Theorem 30 Klini. *The set $T \subseteq \Sigma^*$ is a regular language if and only if it coincides with the language of a classical transducer over Σ^*.*

The technique used for proofing this theorem is the same as this in the classical theorem of Kleene. This is way the proof is omitted.

Let $\mathcal{M} = \langle M, \circ, \mathbf{e} \rangle$ is a monoid. We will proof the following equivalence theorem about co-regular languages.

Theorem 31. *The set $L \subseteq M$ is a co-regular language, if and only if there exists a monoid transducer $\mathcal{T}$, such that $L = L(\mathcal{T})$.*

Proof.

($\Longrightarrow$) Let $L \subseteq M$ is a co-regular language. Then from Theorem 25 there exist a finite alphabet Σ_0, regular language $T \subseteq \Sigma_0^*$ and homomorphism $\varphi : \Sigma_0^* \to M$, such that $\varphi(T) = L$. From Theorem 30 there exists a classical transducer $\mathcal{K}$ over Σ_0^*, such that $T = L(\mathcal{K})$. Then from Theorem 28 there exists a monoid transducer $\mathcal{T}$, such that $\varphi(L(\mathcal{K})) = L(\mathcal{T})$. In this way we obtain: $L = \varphi(T) = \varphi(L(\mathcal{K})) = L(\mathcal{T})$.

($\Longleftarrow$) Let $\mathcal{T}$ is a monoid transducer over $\mathcal{M}$. Then from Theorem 29 there exist a finite alphabet Σ_0, classical transducer $\mathcal{K}$ and homomorphism $\varphi : \Sigma_0^* \to M$, such that $\varphi(L(\mathcal{K})) = L(\mathcal{T})$. From Theorem 30 there exists a regular language $T \subseteq \Sigma_0^*$, such that $L(\mathcal{K}) = T$. From Theorem 24 there exists a co-regular language $L \subseteq M$, such that $\varphi(T) = L$. In this way we obtain: $L(\mathcal{T}) = \varphi(L(\mathcal{K})) = \varphi(T) = L$. □

Remark. Similar results are described in [Br84].

4 Basic Properties of the Co-Regular Languages

Let we concern us on the monoid $\mathcal{T}(X, F)$. Let $T \subseteq \mathcal{T}(X, F)$ is a co-regular language and $\mathcal{T} = \langle S, s_0, S', \lambda, \mu \rangle$ is the monoid transducer (program machine) over $\mathcal{T}(X, F)$ such that $T = L(\mathcal{T})$.

Proposition 32. *Let $T \subseteq \mathcal{T}(X, F)$ is a co-regular language. Then there exist $X' \subseteq X, F' \subseteq F$, such that $|X'| < \infty, |F'| < \infty$ and for all $\tau \in T$ and $x \in X$ when $\tau(x) \neq x$ follows that*

$$x \in X' \ \&\ \mathrm{var}(\tau(x)) \subseteq X' \ \&\ \mathrm{fun}(\tau(x)) \subseteq F'$$

where $\mathrm{var}(t)$ *is the set of all variables contained in the term t and* $\mathrm{fun}(t)$ *is the set of all functional symbols contained in the term t.*

Proof. The proof is quite simple. The main idea is that only a finite set of functional symbols can be used in the definition of co-regular languages and only a finite set of variables is essential. □

This proposition allows us without loose of generality to assume that $|X| < \infty$ and $|F| < \infty$.

Proposition 33. *Let $\mathcal{T}$ is a transducer machine over $\mathcal{T}(X, F)$. Then the co-regular language $L(\mathcal{T}) = \emptyset$ if and only if $A(\mathcal{T}) = \emptyset$.*

Proposition 34. *The problem of emptiness of a co-regular language is decidable.*

Proof. The proposition follows directly from Proposition 33, the remark that $A(\mathcal{T})$ is a regular language and the classical fact that the problem of emptyness of regular languages is decidable. □

The following result is a generalization of the classical *uvw* Theorem.

Theorem 35. *Let $\mathcal{T} = \langle S, s_0, S', \lambda, \mu \rangle$ is a monoid transducer, $\tau \in L(\mathcal{T}), \tau = \lambda(s_0, \alpha)$ and $|\alpha| > |S|$. Then there exist $u, v, w \in \Sigma^*$, such that $\alpha = uvw, v \neq \varepsilon, |uv| < |S|$ and $\lambda(s_0, uv^i w) \in L(\mathcal{T})$ for all $i \in \mathbb{N}$.*

Proof. From the classical uvw Theorem we obtain that, there exist $u, v, w \in \Sigma^*$, such that $\alpha = uvw, v \neq \varepsilon, |uv| < |S|$ and $uv^iw \in A(\mathcal{T})$. This implies directly that: $\lambda(s_0, uv^iw) \in L(\mathcal{T})$ for $i \in \mathbb{N}$. □

The following algorithm decides whether $\tau \in T$, where $\tau \in \mathcal{T}(X, F)$ and $T \subseteq \mathcal{T}(X, F)$ is a co-regular set defined with the monoid transducer $\mathcal{T} = \langle S, s_0, S', \lambda, \mu \rangle$. (That maens that $T = L(\mathcal{T})$.)

Algorithm 36. Let $M_1, M_2, \ldots \subseteq \mathcal{T}(X, F) \times S$.

Step 1 Let $M_1 := \{\langle \mathbf{id}, s \rangle \mid s \in S'\}, k := 1$.
Step 2 Let $k := k + 1, M_k := \emptyset$.
    For every $\langle \sigma, s \rangle \in M_{k-1}$ we check:
        for every $s' \in S$ and $a \in \Sigma$, such that $\mu(s', a) = s$, check:
            if $\lambda(s', a) \circ \sigma \leq \tau$ and $\langle \lambda(s', a) \circ \sigma, s' \rangle \notin M_1 \cup M_2 \cup \ldots M_k$
            then $M_k := M_k \cup \langle \lambda(s', a) \circ \sigma, s' \rangle$;
                if $\lambda(s', a) \circ \sigma = \tau$ and $s' = s_0$
                then terminate with positive result
Step 3 if $M_k = \emptyset$ then terminate with negative result
    else go to step 2.

Theorem 37. *The Algorithm 36 is always terminating and is terminating with positive result if and only if $\tau \in L(\mathcal{T})$.*

Proof. Analyzing Step 2 it is clear that $M_i \subseteq \{\sigma \mid \sigma \leq \tau\} \times S$. Using Propositions 32 and 18, we obtain that $\left|\{\sigma \mid \sigma \leq \tau\} \times S\right| < \infty$. Also we have that $M_k \cap (M_1 \cup M_2 \cup \ldots \cup M_{k-1}) = \emptyset$. Hence there exists some k, such that, $M_k = \emptyset$. Hence Step 3 is ensuring us in the termination of the algorithm.

Let $t \in T$, then there exist shortest $\alpha \in \Sigma^*$, such that $\tau = \lambda(s_0, \alpha)$ and $\mu(s_0, \alpha) \in S'$. If $\alpha = a_1 a_2 \ldots a_{m-1}$, then there exist $\langle \tau_1, s_1 \rangle, \langle \tau_2, s_2 \rangle, \ldots \langle \tau_m, s_m \rangle$, such that $\tau = \tau_1, s_1 = s_0, \tau_m = \mathbf{id}, s_m \in S'$ and $s_{i+1} = \mu(s_i, a_i), \tau_i = \lambda(s_i, a_i) \circ \tau_{i+1}$ for $i = 1, 2, \ldots, m-1$. Then it is clear that $\langle \tau_m, s_m \rangle \in M_1, \langle \tau_{m-1}, s_{m-1} \rangle \in M_2, \ldots, \langle \tau_1, s_1 \rangle \in M_m$, because the opposite would contradict with the condition that α is the shortest word generating τ. Hence the algorithm can't terminate in Step 3, because $M_1 \neq \emptyset, \ldots, M_m \neq \emptyset$. Hence by $k = m$ the algorithm will terminate with positive result because $\langle \tau, s_0 \rangle \in M_m$.

The opposite direction: let the algorithm is terminating with positive result when $k = m$. Then $\langle \tau, s_0 \rangle \in M_m$ and there exists a word $\alpha \in \Sigma^*, \alpha = a_0 a_2 \ldots a_{m-2}$, such that:

$$
\begin{array}{lll}
\langle \tau_1, s_1 \rangle \in M_{m-1} & s_1 = \mu(s_0, a_0) & \tau = \lambda(s_0, a_0) \circ \tau_1 \\
\langle \tau_2, s_2 \rangle \in M_{m-2} & s_2 = \mu(s_1, a_1) & \tau_1 = \lambda(s_1, a_1) \circ \tau_2 \\
\vdots & \vdots & \vdots \\
\langle \tau_{m-2}, s_{m-2} \rangle \in M_2 & s_{m-2} = \mu(s_{m-3}, a_{m-3}) & \tau_{m-3} = \lambda(s_{m-3}, a_{m-3}) \circ \tau_{m-2} \\
\langle \tau_{m-1}, s_{m-1} \rangle \in M_1 & s_{m-1} = \mu(s_{m-2}, a_{m-2}) & \tau_{m-2} = \lambda(s_{m-2}, a_{m-2}) \circ \tau_{m-1}
\end{array}
$$

where $\tau_{m-1} = \mathbf{id}, s_{m-1} \in S'$. Hence $\tau = \lambda(s_0, \alpha)$ and $\mu(s_0, \alpha) = s_{m-1} \in S'$, what means that $\tau \in T$. □

Definition 38. Let $\mathcal{T} = \langle S, s_0, S', \lambda, \mu \rangle$ is a monoid transducer over $\mathcal{T}(X, F)$.

1. The set of accessible steps substitutions and positions in k steps:

$$D_k(\mathcal{T}) = \{\langle \lambda(s_0, \alpha), \mu(s_0, \alpha) \rangle \in \mathcal{T}(X, F) \times S \mid \alpha \in \Sigma^* \ \& \ |\alpha| < k\}$$

2. The set of accessible substitutions and positions:

$$D(\mathcal{T}) = \bigcup_{k=0}^{\infty} D_k(\mathcal{T})$$

3. Let $D \subseteq \mathcal{T}(X, F) \times S$, then the set of accessible from D substitutions and positions in one step is:

$$N_{\mathcal{T}}(D) = \{\langle \tau \circ \lambda(s, a), \mu(s, a) \rangle \in \mathcal{T}(X, F) \times S \mid \langle \tau, s \rangle \in D, a \in \Sigma\}$$

4. The set of generated substitutions for k steps is:

$$L_k(\mathcal{T}) = \{\tau \in \mathcal{T}(X, F) \,|\, \exists \alpha \in \Sigma^*(|\alpha| < k \,\&\, \mu(s_0, \alpha) \in S' \,\&\, \lambda(s_0, \alpha) = \tau)\}$$

Definition 39. We define depth of the monoid transducer $\mathcal{T}$ — $d(\mathcal{T})$:

$$d(\mathcal{T}) = \max\{d(\lambda(s, a)) \,|\, s \in S, a \in \Sigma\}$$

The following propositions we obtain directly from the definitions.

Proposition 40.

1. $L(\mathcal{T}) = \{\tau \in \mathcal{T}(X, F) \,|\, \exists s \in S'(\langle \tau, s\rangle \in D(\mathcal{T}))\}$.
2. $D_{k+1}(\mathcal{T}) = N_{\mathcal{T}}(D_k(\mathcal{T})) \cup D_k(\mathcal{T})$.
3. $L(\mathcal{T}) = \bigcup_{k=0}^{\infty} L_k(\mathcal{T})$.
4. $L_k(\mathcal{T}) = \{\tau \in \mathcal{T}(X, F) \,|\, \exists \langle \tau, s\rangle \in D_k(\mathcal{T})(s \in S')\}$.
5. $N_{\mathcal{T}}(D_k(\mathcal{T})) = D_k(\mathcal{T}) \Longrightarrow D_k(\mathcal{T}) = D(\mathcal{T})$.
6. $|D_k(\mathcal{T})| < \infty, |L_k(\mathcal{T})| < \infty$.
7. *Let* $L_m(\mathcal{T}) = \emptyset$*, where* $m = |S|$*. Then* $L(\mathcal{T}) = \emptyset$.
8. *Let* $\tau \in L_k(\mathcal{T})$*. Then* $d(\tau) < k.d(\mathcal{T})$.

Remark. All results in this section except Proposition 32, Algorithm 36, Theorem 37 and Proposition 40 Point 7 hold for monoid transducers over any monoid.

5 Unification of Co-Regular Languages

We are interested now only in co-regular languages over the term substitution monoid $\mathcal{T}(X, F)$, where (without loose of generality) we assume that $|X| < \infty$ and $|F| < \infty$.

Theorem 41. *The problem of unification and weak unification of co-regular languages is semi-decidable.*

Proof. Let T_1 and T_2 are two co-regular languages defined with the monoid transducers $\mathcal{T}_1$ and $\mathcal{T}_2$ ($T_1 = L(\mathcal{T}_1)$ and $T_2 = L(\mathcal{T}_2)$). Let T_1 and T_2 are unifiable (weak unifiable). That means that there exist $\tau_1 \in L(\mathcal{T}_1)$ and $\tau_2 \in L(\mathcal{T}_2)$, such that τ_1 and τ_2 are unifiable (weak unifiable). Using Propositions 40 we obtain that there exists $k \in \mathbb{N}$, such that $\tau_1 \in L_k(\mathcal{T}_1)$ and $\tau_2 \in L_k(\mathcal{T}_2)$. It is easy to see that the sets $L_i(\mathcal{T}_j)$ are algorithmic generatible. Hence we can generate successively for $i = 1, 2, \ldots, n, \ldots$ the sets $L_i(\mathcal{T}_j)$ (which are finite sets) and search for $\tau_1 \in L_i(\mathcal{T}_1)$ and $\tau_2 \in L_i(\mathcal{T}_2)$ which are unifiable (weak unifiable). This procedure leads to an algorithm which terminates with a positive result whenever the languages are unifiable. □

The decidability of unification (weak unification) of co-regular languages is still an open problem. That is why we present bellow two sufficient conditions for non-unifiabality of co-regular languages which are effective calculable.

Definition 42. We define the basic set of a co-regular language T — $B(T)$ by induction on the construction of T:

1. $T = \emptyset \Longrightarrow B(T) = \emptyset$.
2. $T = \{\tau\} \Longrightarrow B(T) = \{\tau\}$.
3. $T = T_1 \cup T_2 \Longrightarrow B(T) = B(T_1) \cup B(T_2)$.
4. $T = T_1 \circ T_2 \Longrightarrow B(T) = B(T_2)$.
5. $T = T_1^* \Longrightarrow B(T) = B(T_1) \cup \{\mathbf{id}\}$.

Proposition 43. *Let* T *is a co-regular language.*

1. *Let* $\tau \in T$*, then there exist* $\sigma \in B(T)$*, such that* $\sigma \leq \tau$.
2. $|B(T)| < \infty$.

Proposition 44. *Let the co-regular languages T_1 and T_2 are unifiable (weak unifiable). Then there exist $\sigma_1 \in B(T_1), \sigma_2 \in B(T_2)$ and $\lambda_1, \lambda_2 \in \mathcal{T}(X, F)$, such that:*

1. $\lambda_1 \circ \sigma_1$ *and* $\lambda_2 \circ \sigma_2$ *are unifiable (weak unifiable).*
2. $\lambda_1 \circ \sigma_1 \in T1, \lambda_2 \circ \sigma_2 \in T2.$

The propositions given above are proofed straight forward by the definitions.

Theorem 45. *Let T_1 and T_2 are two co-regular languages such that every two substitutions $\tau_1 \in B(T_1)$ and $\tau_2 \in B(T_2)$ are not* **weak** *unifiable. Then T_1 and T_2 are not weak unifiable.*

Proof. This theorem follows dirrectly from Propositions 43 and 44. □

Definition 46. The set of output substitutions for the monoid transducer $\mathcal{T} = \langle S, s_0, S', \lambda, \mu \rangle$ over $\mathcal{T}(X, F)$ we define as follows:

$$O(\mathcal{T}) = \{\lambda(s, a) \in \mathcal{T}(X, F) \,|\, \exists a \in \Sigma, s \in S(\mu(s, a) \in S')\}.$$

Theorem 47. *Let $\mathcal{T}_1 = \langle S_1, s_{01}, S_1', \lambda_1, \mu_1 \rangle$ and $\mathcal{T}_2 = \langle S_2, s_{02}, S_2', \lambda_2, \mu_2 \rangle$ are monoid transducers over $\mathcal{T}(X, F)$, such that every two substitutions $\tau_1 \in O(\mathcal{T}_1)$ and $\tau_2 \in O(\mathcal{T}_2)$ are not* **weak** *unifiable, and $s_{01} \notin S_1', s_{02} \notin S_2'$. Then the co-regular languages $L(\mathcal{T}_1)$ and $L(\mathcal{T}_2)$ are not weak unifiable.*

Proof. Let $L(\mathcal{T}_1)$ and $L(\mathcal{T}_2)$ are (weak) unifiable. Hence there exist $\tau_1 \in L(\mathcal{T}_1)$, $\tau_2 \in L(\mathcal{T}_2)$ which are (weak) unifiable. Then $\tau_i = \lambda_i(s_{0i}, \alpha_i)$ and $|\alpha_i| > 1$ (because $s_{0i} \notin S_i'$). Let $\alpha_i = \alpha_i' a_i$, where $a_i \in \Sigma$. Then $\tau_i = \lambda_i(s_{0i}, \alpha_i') \circ \lambda_i(\mu_i(s_{0i}, \alpha_i'), a_i)$ and $\mu_i(\mu_i(s_{0i}, \alpha_i'), a_i) \in S_i'$. Hence $\lambda_i(\mu_i(s_{0i}, \alpha_i'), a_i) \in O(\mathcal{T}_i)$, and $\lambda_1(\mu_1(s_{01}, \alpha_1'), a_1)$ and $\lambda_2(\mu_2(s_{02}, \alpha_2'), a_2)$ are weak unifiable which is contradiction with the conditions. □

6 Conclusion

This work presents many open problems. The most important (in our opinion) is the decidability of the co-regular languages unification. Bellow there is a list of other open problems. Notice that we are interested especially on the monoid $\mathcal{T}(X, F)$.

Problem 48. Is the problem of equivalence of co-regular languages decidable?

Remark. In the case of regular languages this problem is decidable. The technique essential uses the fact that the complement of a regular language is a regular language too. In the case of co-regular languages over $\mathcal{T}(X, F)$ this is not true. Hence it seems that this problem is not decidable.

Problem 49. Let $\mathcal{T} = \langle S, s_0, S', \lambda, \mu \rangle$ is a monoid transducer. If there exists $k \in \mathbb{N}$, such that $L_k(\mathcal{T}) = L_{k+m}(\mathcal{T})$, then $L(\mathcal{T}) = L_k(\mathcal{T})$, where $m = |S|$. Is there any limitation from above for k?

Remark. This is a quite natural hypothesis. If this result holds and there exists an effective calculable limit for k it would lead to a procedure for detection and generation of finite co-regular languages.

Problem 50. Are there any necessary and/or sufficient conditions for the existence of such $k \in \mathbb{N}$, that for the given co-regular language $T \subseteq \mathcal{T}(X, F)$, holds $T^k = T^*$.

References

[Br84] Wilfried Brauer: *Automatentheorie.* B. G. Teubner Stuttgart, 1984.

[Bu78] Anatoli Buda: *Abstract Program Machines.* pre-print 108, Academy of Sciences USSR, Novosibirsk, 1978. *in Russian,*

[Ed85] Elmar Eder: *Properties of Substitutions and Unifications.* Symbolic Computation 1, 31-46, 1985.

[Mi93] Stoyan Mihov: *Unification of Co-regular Languages.* M. Sc. Thesis at the Faculty of Mathematics, Sofia University, 1993. *in Bulgarian.*

[Ro65] J. A. Robinson, *A machine-oriented logic based on the resolution principle.* J. Assoc. Comp. Mach. 12, 23-41, 1965.

This article was processed using the LaTeX macro package with LLNCS style