Acta Cybernetica **22** (2016) 613–616.

A Note on the Emptiness of Intersection Problem for Left Szilard Languages

Erkki Mäkinen*

Abstract

As left Szilard languages form a subclass of simple deterministic languages and even a subclass of super-deterministic languages, we know that their equivalence problem is decidable. In this note we show that their emptiness of intersection problem is undecidable. The proof follows the lines of the correponding proof for simple deterministic languages, but some technical tricks are needed. This result sharpens the borderline between decidable and undecidable problems in formal language theory.

Keywords: left Szilard languages, Post Correspondence Problem, emptiness of intersection

1 Introduction

Let $G = (N, T, P, S)$ be a context-free grammar where N is the alphabet of nonterminals, T is the alphabet of terminals, P is the set of productions, and S is the start symbol. Suppose that each production in P has the form $A \to a\alpha$ where $a \in T$ and $\alpha \in N^*$. Now, if $A \to a\alpha$ and $B \to a\beta$ in P always implies $A = B$ and $\alpha = \beta$ (that is, the right hand sides start with unique terminals), we say that the grammar is a *left Szilard grammar* and the language generated is a *left Szilard language* [5]. Left Szilard languages are also known as *very simple languages* [6].

As left Szilard languages are simple deterministic languages (in the sense of Korenjak and Hoproft [4]) and super-deterministic languages (in the sense of Greibach and Friedman [1]), their equivalence problem is decidable. On the other hand "$L = L_1$?" is undecidable for a context-free language L and a left Szilard language L_1, since there are unbounded left Szilard languages, which makes the problem undecidable [3, 1].

An instance of *Post correspondence problem* (PCP) consists of two lists of words $(w_1, w_2, \ldots, w_n)$ and $(y_1, y_2, \ldots, y_n)$ over an alphabet Σ. A *solution* is a non-empty sequence of indices $i_1, \ldots, i_k$ such that $w_{i_1} \ldots w_{i_k} = y_{i_1} \ldots y_{i_k}$. It is undecidable whether such a solution exists or not for a given instance of PCP [2].

*School of Information Sciences, FI-33014 University of Tampere, Finland E-mail: {Erkki.Makinen}@uta.fi

DOI: 10.14232/actacyb.22.3.2016.4

The standard procedure to start considering undecidability problems for formal languages is to reduce PCP to the emptiness of intersection problem for context-free languages. This reduction is possible also for simple deterministic languages [4] and for super-deterministic languages [1]. This note shows that the reduction is possible also to the emptiness of intersection problem for the left Szilard languages.

2 The result

Consider an instance of PCP with lists $(w_1, w_2, \ldots, w_n)$ and $(y_1, y_2, \ldots, y_n)$ over an alphabet Σ. The text book proof (see, e.g., [2]) for the undecidability of the empiness of intersection problem for context-free languages uses grammars with productions $A \to w_1Aa_1 \mid \cdots \mid w_nAa_n \mid w_1a_1 \mid \cdots \mid w_na_n$ and $B \to y_1Ba_1 \mid \cdots \mid y_nBa_n \mid y_1a_1 \mid \cdots \mid y_na_n$, where a_i's are the unique labels of the words in the w-list and y-list. In order to transform the productions into the correct left Szilard form, we first change the places of w_i's and a_i's (resp. y_i's and a_i's), so that the unique indices can be interpreted as the unique terminals required to be in the begining of the right hand sides of the productions in a left Szilard grammar. Simultaneously, we take the mirror image of each w_i (resp. y_i) in order to keep the letters in the correct order in the resulting sentence (from right to left). Hence, if $A \to w_{i_1} \ldots w_{i_k}Aa_i$ (resp. $B \to y_{i_1} \ldots y_{i_k}Ba_i$) is a production in the original grammar, we change it to be $A \to a_iAw_{i_k} \ldots w_{i_1}$ (resp. $B \to a_iB \to y_{i_k} \ldots y_{i_1}$), or by using the standard notation for mirror image, we change $A \to w_iAa_i$ (resp. $B \to y_iAa_i$) to be $A \to a_iAw_i^{-1}$ (resp. $B \to a_iBy_i^{-1}$).

Both the set of A-productions and the set of B-productions constructed above contain now exactly two productions with their right hand sides starting with each of the indices a_i. The productions of the form $A \to a_iw_i$ (resp. $B \to a_iy_i$) are applied only once (as the last production) in each derivation resulting a terminal word. Therefore, we can replace each production $A \to a_iw_i$ (resp. $B \to a_iy_i$) by a production $A \to \delta_iw_i^{-1}$ (resp. $B \to \delta_iy_i^{-1}$) where δ_i's are new terminal symbols over an alphabet Δ. Notice that mirror images are needed also in these productions.

Moreover, for each symbol x in Σ, we add X, where X is a new symbol, to the set of nonterminals and the production $X \to x$ to the set of productions. Each $x \in \Sigma$ in the productions so far produced is replaced with X. All the productions are now of the required form with unique terminals in the beginning of their right hand sides.

Next we formally define the left Szilard grammars to which a given instance of PCP is reduced. Let the instance consist of the lists $W = (w_1, \ldots, w_n)$ and $Y = (y_1, \ldots, y_n)$ over Σ. Define a left Szilard grammar G_W as $(\{A\} \cup X_\Sigma, \Sigma \cup I \cup \Delta, P_W, A)$ where $X_\Sigma = \{X_{a_i} \mid a_i \in \Sigma\}$, $I = \{a_i \mid i = 1, \ldots, n\}$, $\Delta = \{\delta_i \mid i = 1, \ldots, n\}$ and P_W contains the productions $A \to a_iAw_i^{-1}$ and $A \to \delta_iw_i^{-1}$, for each w_i in the list W, and the production $X_{a_i} \to a_i$, for each $a_i \in \Sigma$. Similarly, define a left Szilard grammar G_Y as $(\{B\} \cup X_\Sigma, \Sigma \cup I \cup \Delta, P_Y, B)$ where X_Σ, I, and Δ are as in G_W, and P_Y contains the productions $B \to a_iBy_i^{-1}$ and $B \to \delta_iy_i^{-1}$, for each y_i in the list Y, and the production $X_{a_i} \to a_i$, for each $a_i \in \Sigma$.

If the PCP instance has a solution $i_1, \ldots, i_k$, we have $w_{i_1} \ldots w_{i_k} = y_{i_1} \ldots y_{i_k}$. Clearly, this happens if and only if the intersection $L(G_W) \cap L(G_Y)$ contains the word $a_{i_1} \ldots a_{i_{k-1}} \delta_{i_k} w_{i_k}^{-1} \ldots w_{i_1}^{-1} = a_{i_1} \ldots a_{i_{k-1}} \delta_{i_k} y_{i_k}^{-1} \ldots y_{i_1}^{-1}$.

We have proved the following theorem.

Theorem 1. *The emptiness of intersection problem is undecidable for left Szilard languages.*

We end this chapter by an example of the above construction. Let the lists $W = (a, abaaa, ab)$ and $Y = (aaa, ab, b)$ form an instance of PCP. The words in the lists contain letters a and b; hence, we have $\Sigma = \{a, b\}$. The sequence of indices $2-1-1-3$ is a solution for this instance and the common string corresponding to these indices is aba^6b. The corresponding left Szilard grammar G_W has the productions $A \to 1AX_a$, $A \to 1_\delta X_a$, $A \to 2AX_aX_aX_aX_bX_a$, $A \to 2_\delta X_aX_aX_aX_bX_a$, $A \to 3AX_bX_a$, $A \to 3_\delta X_bX_a$, $X_a \to a$, and $X_b \to b$. Similarly, the left Szilard grammar G_Y has the productions $B \to 1BX_aX_aX_a$, $B \to 1_\delta X_aX_aX_a$, $B \to 2BX_bX_a$, $B \to 2_\delta X_bX_a$, $B \to 3BX_b$, $B \to 3_\delta X_b$, $X_a \to a$, and $X_b \to b$.

The word corresponding to $2-1-1-3$ can be generated in G_W and G_Y as follows:

$$\begin{aligned} &A \Rightarrow 2AX_a^3X_bX_a \Rightarrow 21AX_a^4X_bX_a \Rightarrow 211AX_a^5X_bX_a \\ &\Rightarrow 2113_\delta X_bX_a^6X_bbX_a \Rightarrow^+ 2113_\delta ba^6ba \end{aligned}$$

and

$$\begin{aligned} &B \Rightarrow 2BX_bX_a \Rightarrow 21BX_a^3X_bX_a \Rightarrow 211BX_a^6X_bX_a \\ &\Rightarrow 2113_\delta X_bX_a^6X_bX_a \Rightarrow^+ 2113_\delta ba^6ba. \end{aligned}$$

3 Discussion

The emptiness of intersection problem for context-free languages is the basic undecidable problem in formal language theory, as in most treatments it transmits the undecidability of Turing machine computations to language theory. A natural question then is to find the simplest class of languages for which this transmission is possible. Previously, the classes of simple deterministic languages and superdeterministic languages have been known to be enough for the reduction. This note shows that the structure of PCP can be presented even in the terms of left Szilard languages.

References

[1] Greibach, S.A., and Friedman, E.P., Superdeterministic PDAs: A subcase with a decidable inclusion problem. *Journal of the ACM* 27(4):675–700, 1980.

[2] Hopcroft, J.E., Motwani, R., and Ullman, J.D. *Introduction to Automata Theory, Languages, and Computation, 2nd Edition.* Addison-Wesley, 2001.

[3] Hunt III, H.B., and Rangel, J.L. Decidability of equivalence, containment, intersection, and separability of context-free languages. In *Proc. 16th Annual Symposium on Foundations of Computer Science*, pages 144-149, 1975.

[4] Korenjak, A.J., and Hopcroft, J.E. Simple deterministic languages. In *IEEE Conference Record of Seventh Annual Symposium on Switching and Automata Theory*, pages 36-46, 1966.

[5] Mäkinen, E. On context-free derivations. *Acta Universitatis Tamperensis* Ser. A, Vol. 198, 1985.

[6] Yokomori, T. Polynomial-time identification of very simple grammars from positive data. *Theoretical Computer Science* 298,1:179-206, 2003.

Received 19th August 2015