Post's Correspondence Problem and the Undecidability of Context-Free Intersection

Wim H. Hesselink

July 28, 2015

1 Introduction

Our starting point is the undecidability of nullability of a string in a rewrite system. This is used to prove undecidability of Post's Modified Correspondence Problem, and of the disjointness of context-free languages. These results are not new but the presentation differs from the book [HU79].

In courses on languages, one usually omits the operator for concatenation of strings or concatenation of elements to strings or strings to elements. Strings are finite sequences of symbols, but in this note we also need to concatenate sequences of pairs. We here therefore decided to use the operator : to concatenate finite sequences, and elements to finite sequences, and finite sequences to elements. We also use them for strings, but we omit them when they become confusing, in particular, between concrete symbols.

An *alphabet* is a finite set; its elements are called *symbols*. A *string* over alphabet Σ is a finite sequence of elements of Σ. The empty string is denoted by ε. For a string x and a symbol a, we write $n_a(x)$ to denote the number of occurrences of symbol a in sequence x. The reversal of a string x is denoted by x^R. Recall that $(x : y)^R = y^R : x^R$ for arbitrary strings x and y.

2 Rewrite systems and nullability

A *rewrite system* is a finite set P of pairs of strings over an alphabet Σ.

Given a rewrite system P, a pair $(u, v) \in P$ is called a *rewrite rule*; it is usually written $u \to v$. The step relation $\Rightarrow_P$ between strings x and y is defined by

$$x \Rightarrow_P y \quad \equiv \quad (\exists\, u, v, w, z \in \Sigma^* : x = u : v : w \ \wedge\ (v, z) \in P \ \wedge\ u : z : w = y) \ .$$

The derivation relation $\Rightarrow_P^*$ is the reflexive transitive closure of $\Rightarrow_P$.

The nullability problem NULB is the question whether a string has a derivation to the empty string. Formally, it is defined as follows:

Problem NULB. Instance: a pair (P, w) formed by rewrite system P and string w. Question: $w \Rightarrow_P^* \varepsilon$?

It is known that problem NULB is undecidable. This can be proved by reducing Turing's HALTING problem to NULB.

3 Post's Correspondence Problem

For any set X, the set X^* is the set of the finite sequences of elements of X. For a function $f : X \to Y$, the induced function is denoted $f^* : X^* \to Y^*$. Let π_1 and

π_2 be the two projection functions from a Cartesian product $X \times X$ to X given by $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$.

Let Σ be an alphabet. As Σ^* is the set of the strings over Σ, Σ^{**} is the set of the finite sequences of strings over Σ. The function $concat : \Sigma^{**} \to \Sigma^*$ is defined to yield the concatenation of the elements of its argument. For $i = 1$, 2, let $cc_i : (\Sigma^* \times \Sigma^*)^* \to \Sigma^*$ denote the composition $cc_i = concat \circ \pi_i^*$.

Post's Correspondence Problem (PCP) was introduced in [Pos46]. It can be formulated as follows.

Problem PCP. Instance: a finite set U of pairs of strings.
Question: does there exist a nonempty sequence $us \in U^*$ with $cc_1(us) = cc_2(us)$?

Example. Let U contain the pairs $u_1 = (dog, doge)$ and $u_2 = (eats, at)$, and $u_3 = (print, sprint)$. Then $us = (u_1, u_2, u_3)$ is a solution because both $cc_1(us)$ and $cc_2(us)$ are equal to the string *dogeatsprint*.

The modified Post's Correspondence Problem replaces the nonemptiness of the sequence by supplying a specific starting pair:

Problem MPCP. Instance: a pair (p, V) formed by a pair p of strings and a finite set V of pairs of strings.
Question: does there exist a sequence $vs \in V^*$ with $cc_1(p : vs) = cc_2(p : vs)$?

The following lemma serves to relate problem NULB to problem MPCP:

Lemma 1 *Let P be a rewrite system over Σ. Let x and y be strings over Σ. Put $V_0 = P \cup \{(a, a) \mid a \in \Sigma\}$.*
(a) If $x \Rightarrow_P y$, then there is a sequence $s \in V_0^*$ with $x = cc_1(s)$ and $y = cc_2(s)$.
(b) If there is a sequence $s \in V_0^*$ with $x = cc_1(s)$ and $y = cc_2(s)$, then $x \Rightarrow_P^* y$.

Proof. (a) By definition, there are strings u, v, w, z with $x = u : v : w$ and $y = u : z : w$ and $(v, z) \in P$. Let $\Delta : \Sigma \to V_0$ be given by $\Delta(a) = (a, a)$ for all $a \in \Sigma$. Then $s = \Delta^*(u) : (v, z) : \Delta^*(w)$ satisfies the requirements.

(b) Sequence s may contain several elements of P, and a number of pairs of the form (a, a) with $a \in \Sigma$. The elements of P act as independent rewrite rules, and can be applied in arbitrary order. □

Theorem 2 *Problem MPCP is not decidable.*

Proof. This is proved by reduction of NULB to MPCP. Let (P, w) be an instance of NULB. We convert this instance of NULB to an instance of MPCP, which is positive if and only if the instance of NULB is positive. Any decision procedure for MPCP can therefore be translated to a decision procedure for NULB. As problem NULB is undecidable, it will follow that MPCP is undecidable.

We thus start with the instance (P, w) of NULB. Let Σ be the finite set of the symbols that occur in P and w. Let V_0 be the set of pairs introduced in Lemma 1. We can choose a symbol $d \notin \Sigma$, and consider the instance (p, V) of MPCP given by

$$p = (d, d : w : d) \text{ and } V = V_0 \cup \{(d, d), (dd, d)\} .$$

Assume that vs is a solution of the instance (p, V) of MPCP. Let $xs = cc_1(p : vs)$ and $ys = cc_2(p : vs)$, so that $xs = ys$. As d does not occur in the set of rewrite rules V_0, and $n_d(d) = 1$ and $n_d(d : w : d) = 2$, the pair (dd, d) has precisely one occurrence in the sequence vs. We use the occurrences of pair (d, d) to split the prefix of vs before (dd, d) into subsequences $u_1, \ldots, u_k$ separated by pairs (d, d):

$$vs = u_1 : (d, d) : u_2 : (d, d) : \ldots (d, d) : u_k : (dd, d) : \ldots .$$

Now the symbol d does not occur in the sequences u_i for $1 \le i \le k$. We have

$$p : vs = (d, d : w : d) : u_1 : (d, d) : u_2 : (d, d) : \ldots (d, d) : u_k : (dd, d) : \ldots ,$$

Define $x_i = cc_1(u_i)$ and $y_i = cc_2(u_i)$. As $xs = cc_1(p : vs)$ and $ys = cc_2(p : vs)$, it follows that

$$\begin{aligned} xs &= d : x_1 : d : x_2 : \ldots d \ : \ x_k \ : \ d \quad : \quad d : \ldots \\ ys &= d : \ w \ : d : y_1 : \ldots d : y_{k-1} : \ d : y_k : d : \ldots , \end{aligned}$$

where d does not occur in the strings x_i and y_i for $1 \leq i \leq k$. As $ys = xs$, it follows that $w = x_1$, and that $y_i = x_{i+1}$ for $1 \leq i < k$, and that $y_k = \varepsilon$.

For $1 \leq i \leq k$, the symbol d does not occur in u_i. Therefore, $u_i \in V_0^*$. As $x_i = cc_1(u_i)$ and $y_i = cc_2(u_i)$, Lemma 1(b) implies $x_i \Rightarrow_P^* y_i$ for $1 \leq i \leq k$.

Combined with the previous paragraph, this yields

$$w = x_1 \Rightarrow_P^* x_2 \Rightarrow_P^* \ \ldots \ \Rightarrow_P^* x_k \Rightarrow_P^* \varepsilon \ .$$

This proves that (P, w) is a positive instance of problem NULB.

Conversely, assume that (P, w) is a positive instance of NULB. Then there is a derivation

$$w = x_1 \Rightarrow_P x_2 \Rightarrow_P \ \ldots \ \Rightarrow_P x_k \Rightarrow_P x_{k+1} = \varepsilon \ .$$

By Lemma 1(a), there are sequences $u_i \in V_0^*$ $(1 \leq i \leq k)$ such that $x_i = cc_1(u_i)$ and $x_{i+1} = cc_2(u_i)$. It follows that the sequence

$$vs = u_1 : (d, d) : u_2 : (d, d) : \ldots (d, d) : u_k : (dd, d)$$

solves the instance of MPCP.

This concludes the proof that the instance (P, w) of NULB is positive if and only if its translation (p, V) is a positive instance of MPCP. As NULB is undecidable, this concludes the proof that problem MPCP is undecidable. □

Remark. In [HU79], this result is used to prove the undecidability of PCP. □

4 Postian grammars

We define a context-free grammar G over the terminal alphabet Σ to be *Postian* iff its start symbol S is its only nonterminal, the grammar has one production rule $S \to u$ with $n_S(u) = 0$ and $n_e(u) = 1$ for some terminal symbol $e \in \Sigma$, and all other production rules $S \to u$ satisfy $n_S(u) = 1$ and $n_e(u) = 0$.

Problem PALIN. Instance: a Postian grammar G.
Question: does $L(G)$ contain a palindrome?

Lemma 3 *Problem PALIN is equivalent to MPCP.*

Proof. We show that every instance (p, V) of MPCP can be converted to an instance of PALIN that has a solution if and only if the instance of MPCP has a solution, and that, moreover, every instance of PALIN is obtained in this way.

Let Σ be the set of the symbols that occur in (p, V). Let e be a symbol not in Σ. Write $p = (p_1, p_2)$. Consider the Postian grammar

$$(0) \qquad G: \quad \begin{aligned} S &\to x^R : S : y \quad \text{for all } (x, y) \in V \\ S &\to p_1^R : e : p_2 \ . \end{aligned}$$

Every derivation of a string in $L(G)$ consists of a sequence of applications of the rules $S \to x^R : S : y$ with $(x, y) \in V$, concluded by one application of the rule $S \to p_1^R : e : p_2$. Taken in reverse order, this amounts to a sequence of pairs $p : vs$ with $vs \in V^*$, which gives rise to a terminal string

$$F(vs) = (cc_1(p : vs))^R : e : (cc_2(p : vs)) \ .$$

This proves that $L(G)$ consists of the strings $F(vs)$ for all $vs \in V^*$. As $F(vs)$ contains precisely one occurrence of symbol e, it is a palindrome if and only if $cc_1(p : vs)) = cc_2(p : vs)$, i.e., if and only if $(p : vs)$ solves the instance of MPCP.

Conversely, for every Postian grammar G, there is a unique pair (p, V) such that G is of the form (0). □

As MPCP is undecidable, this implies

Theorem 4 *Problem PALIN is undecidable.*

As the set of the palindromes over a given alphabet is context-free, problem PALIN is a special case of the problem whether two context-free languages have a nonempty intersection:

Problem CFI. Instance: two context-free grammars G_1 and G_2.
Question: is the intersection $L(G_1) \cap L(G_2)$ nonempty?

Therefore, the problem CFI is also undecidable.

References

[HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[Pos46] E. Post. A variant of a recursively unsolvable problem. *Bulletin of the AMS*, 52:264–268, 1946.