An Language you cannot prove not regular by Pumping (Allegedly)

Ehrenfeucht, et al \[1\] exhibit, for all languages $Z \subseteq \{1, 2\}^*$ a languages $L_Z$ (the mapping $Z$ goes to $L_Z$ is injective) such that $L_Z$ cannot be proven not regular by the Pumping Lemma (they show this for a rather advanced version of the pumping lemma). Since most of these $L_Z$ are not regular, this would seem show there are many non-regular languages that cannot be proven non-regular by the pumping lemma. In this note we show that, using closure properties and a simple form of the pumping lemma, the languages $L_Z$ that are non-regular can be proven to be non-regular.

**Notation 0.1**

$\Sigma$ is the 16-letter alphabet $\{(i,j) : 0 \leq i,j \leq 3\}$.

$f_1 : \Sigma \to \Sigma$ is defined by

$$f_1((i,j)) = (i + 1 \text{mod } 4, j)$$

$f_2 : \Sigma \to \Sigma$ be defined by

$$f_2((i,j)) = (i, j + 1 \text{mod } 4)$$

Note that $f_1(f_2(\sigma)) \neq f_2(f_1(\sigma))$.

**Def 0.2** A string $x$ is *legal* if

1. $x = (\sigma_1)^{n_1}(\sigma_2)^{n_2} \cdots (\sigma_m)^{n_m}$ where $n_1, n_2, \ldots, m \geq 1$.
2. $\sigma_1 = (0, 0)$.
3. For all $2 \leq i \leq m$, either $\sigma_i = f_1(\sigma_{i-1})$ or $\sigma_i = f_2(\sigma_{i-1})$.

Example:

$$(0,0)(1,0)(1,0)(1,0)(2,0)(2,1)(3,1)(0,1)$$

We associate to every legal string the sequence of transitions that cause $\sigma_i$ to go to $\sigma_{i+1}$, called the code string. Note that above:

- $f_1(0,0) = (1,0)$
- $f_1(1,0) = (2,0)$
- $f_2(2,0) = (2,1)$
- $f_1(2,1) = (3,1)$
- $f_1(3,1) = (0,1)$.

So we associate code string 11211.

Lets go in the other direction: We give legal strings with code string 11211:

$$(0,0)^{\geq 1}(1,0)^{\geq 1}(2,0)^{\geq 1}(2,1)^{\geq 1}(3,1)^{\geq 1}(0,1)^{\geq 1}$$

**Def 0.3** Let $x \in \Sigma^*$. The *parity* of $x$ is the parity of the sum of all of the components of $x$. 

1
Example: The parity of

\[(0,0)(1,0)(1,0)(1,0)(2,0)(2,1)(3,1)(0,1)\]

is

\[0 + 0 + 1 + 0 + 1 + 0 + 2 + 0 + 2 + 1 + 3 + 1 + 0 + 1 \pmod{2} = 1.\]

Def 0.4 Let \(Z \subseteq \{1,2\}^*\). Let

\[L_Z = \{x : x \text{ is legal and } (\exists z \in Z)[x \text{ has code strings } z]\} \cup \{x : x \text{ is not legal and parity}(x)=0\}\]

We leave the following easy theorem to the reader.

Theorem 0.5 If \(Z\) is regular then \(L_Z\) is regular.

Ehrenfeucht, et al [1] prove that, for all \(Z\), \(L_Z\) cannot be proven non-regular using the pumping lemma. Since there are an uncountable number of \(Z\), and each \(Z\) gives a different \(L_Z\), there are an uncountable number of non-regular languages that cannot be proven not-regular by the pumping lemma.

We use closure properties to show that if \(L_Z\) is regular than \(Z\) is regular.

Def 0.6 Let \(\Sigma_1\) and \(\Sigma_2\) be finite alphabets. Let \(F : \Sigma_1 \times \Sigma_1 \rightarrow \Sigma_2\). We extend \(F\), first to \(\Sigma_1^*\), second to all subsets of \(\Sigma_1^*\).

1. Let \(F : \Sigma_1^* \rightarrow \Sigma_2^*\) be defined by

\[F(\sigma_1\sigma_2\sigma_3\cdots\sigma_n) = f(\sigma_1\sigma_2)f(\sigma_2\sigma_3)\cdots f(\sigma_{n-2}\sigma_{n-1})f(\sigma_{n-1}\sigma_n).\]

2. Let \(F : 2^{\Sigma_1^*} \rightarrow 2^{\Sigma_2^*}\) be defined by

\[F(L) = \{f(x) : x \in L\}.\]

Lemma 0.7 Let \(\Sigma_1\) and \(\Sigma_2\) be finite alphabets. Let \(f : \Sigma_1 \times \Sigma_1 \rightarrow \Sigma_2\). Let \(F\) be as in definition 0.6. Let \(L \subset \Sigma_1^*\) such that if \(L\) is regular then \(F(L)\) is regular.

Theorem 0.8 Let \(Z \subseteq \{0,1\}^*\). If \(L_Z\) is regular then \(Z\) is regular.

Proof: Assume \(L = L_Z\) is regular. Note that

\[PAR1 = \{x : x \text{ has parity } 1\}\]

is regular. Hence

\[L' = L \cap PAR1 = \{x : x \text{ is legal and } x \text{ has parity } 1 \text{ and } (\exists z \in Z)[x \text{ has code strings } z]\}\]
is regular.

Let

\[ NOD = \{ x = \sigma_1 \cdots \sigma_n : (\forall i \leq n - 1)[\sigma_i \neq \sigma_{i+1}] \} \]

\((NOD\) stands for NO Doubles.\)

Note that \(NOD\) is regular. Hence \(L' \cap NOD\) is regular. If \(x \in L' \cap NOD\) then the following hold:

1. \(x = \sigma_1 \sigma_2 \cdots \sigma_m\) where, for all \(1 \leq i \leq m - 1, \sigma_i \neq \sigma_{i+1}\).
2. \(\sigma_1 = (0,0)\).
3. For all \(2 \leq i \leq m\), either \(\sigma_i = f_1(\sigma_{i-1})\) or \(\sigma_i = f_2(\sigma_{i-1})\).
4. \(x\) has parity 1.
5. \(x\) codes \(z\).

One can easily construct a DFA for \(Z\) from a DFA for \(L' \cap NOD\). Hence \(Z\) is regular.

\[ \blacksquare \]

References