An almost-confluent congruential language which is not a Church-Rosser language.

Colm Ó Dúnlaing

Trinity College, Dublin, Ireland

The application of string-rewriting systems to formal languages goes back to Nivat [STOC 1970], but the notation and terms have evolved [Jantzen, 1988; Book and Otto, 1993].

We shall discuss Church-Rosser congruential languages (CRCLs) and almost-confluent congruential languages (ACCLs), defined later.

Two difficult problems (in this area!) have been solved in this century.

- Jurdziński and Loryś: palindromes are not a Church-Rosser language [ICALP 2002].
- It was long known that every regular set is an ACCL. Diekert, Kufleitner, Reinhardt, and Walter showed that regular sets are CRCLs [ICALP 2012, JACM 2015].

This left open the question: is every ACCL a CRCL? We provide a counter-example.

A Thue system T is a (finite) set of *rules* through which one string x can be converted to another y by repeatedly replacing substrings mentioned in the rules. Rules of T are written as $u \leftrightarrow v$, meaning that occurrences of u can be replaced by v, *or vice-versa*.

That is, $x \leftrightarrow_T y$ means that y can be obtained from x by replacing a substring u in x by v in y where either $u \leftrightarrow v$ or $v \leftrightarrow u$ is a rule in T.

Write $x \overset{*}{\leftrightarrow}_T y$ if y can be derived from x by a sequence of such operations. This relation is a congruence, i.e.,

- An equivalence relation, which respects concatenation:
- $x_1 \overset{*}{\leftrightarrow}_T x_2 \ \wedge \ y_1 \overset{*}{\leftrightarrow}_T y_2 \implies x_1 y_1 \overset{*}{\leftrightarrow}_T x_2 y_2$

Write

$$x \rightarrow_T y$$

when

$$x \leftrightarrow_T y \ \wedge \ |y| < |x|$$

(i.e., y is shorter than x).

When $x \overset{*}{\rightarrow}_T y$, we say that x *reduces to* y.

The Thue system T is *Church-Rosser* if whenever $x \overset{*}{\leftrightarrow}_T y$, both x and y can be reduced to the same string z. Of course, the string z can be assumed irreducible.

A very simple example of a Church-Rosser Thue system over a binary alphabet $\{a, b\}$ is

$$T = \{ab \rightarrow \lambda\}.$$

(λ is the empty string).

Reducing in this system means cancelling occurrences of ab. The irreducible strings (containing no occurrence of ab) are

$$\lambda,\ a,\ b,\ aa,\ ba,\ bb,\ aaa,\ baa,\ bba,\ bbb\ldots$$

A generalisation of Church-Rosser Thue systems is an almost confluent Thue system. Define

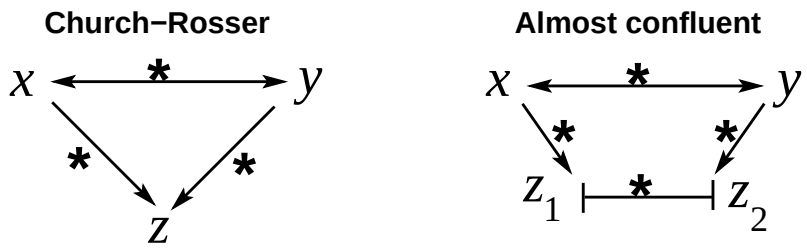
$$x \vdash\!\!\dashv_T y$$

when $x \leftrightarrow_T y$ and $|x| = |y|$.

T is *almost confluent* if $x \overset{*}{\leftrightarrow}_T y \iff$

$$\exists z_1, z_2$$

$$x \overset{*}{\rightarrow}_T z_1 \ \wedge\ z_1 \overset{*}{\vdash\!\!\dashv}_T z_2 \ \wedge\ y \overset{*}{\rightarrow}_T z_2$$



An almost confluent congruential language (ACCL) is a finite union of congruence classes of an almost confluent Thue system.

It is relatively easy to show that regular sets are ACCLs. It is very hard to show that they are CRCLs [Diekert, Kufleitner, Reinhardt, Walter].

An ACCL L which is not a CRCL

L is derived from an unusual presentation of $\mathbb{Z}$. Take a 4-letter alphabet $\Sigma = \{a, b, \overline{a}, \overline{b}\}$. Define

$$\text{POS} = \{a, b\}, \quad \text{NEG} = \{\overline{a}, \overline{b}\}.$$

Definition. Given $x \in \Sigma^*$,

$$|x|_{\text{pos}}, |x|_{\text{neg}}$$

are the number of occurrences in x of strings from POS (respectively, NEG).

Define a map $h : \Sigma^* \to \mathbb{Z}$

$$h(x) = |x|_{\text{pos}} - |x|_{\text{neg}}.$$

Proposition. h is a homomorphism (easy).

The language L:

$$L = h^{-1}(0)$$

First, we shall show that L is an ACCL, by producing an almost confluent Thue system S such that

$$L = [\lambda]_S$$

S will be the following Thue system.

Length-reducing rules:

$$a\overline{a} \to \lambda,\ a\overline{b} \to \lambda,\ b\overline{a} \to \lambda,\ b\overline{b} \to \lambda,$$
$$\overline{a}a \to \lambda,\ \overline{b}a \to \lambda,\ \overline{a}b \to \lambda,\ \overline{b}b \to \lambda$$

Length-preserving rules:

$$a \longmapsto b,\ \overline{a} \longmapsto \overline{b}$$

Note that if $\alpha \to \lambda$ is a length-preserving rule, then $h(\alpha) = h(\lambda) = 0$, and if $\alpha \longmapsto \beta$ is a length-preserving rule, then $h(\alpha) = h(\beta) = \pm 1$. It follows by induction that

$$x \overset{*}{\leftrightarrow}_S y \implies h(x) = h(y).$$

Call strings in POS* (NEG*) positive (respectively, negative). Call the other strings

mixed. Clearly, the mixed strings are reducible (modulo S): the others are irreducible.

Given two strings x, y such that

$$x \overset{*}{\leftrightarrow}_S y$$

Reduce x to an irreducible string z_1 and y to z_2. Then

$h(x) = h(y) = h(z_1) = h(z_2) = \pm|z_1| = \pm|z_2|$. Since z_1 and z_2 are both positive or negative, and have the same length, they can be intraconverted using the two length-preserving rules. **Therefore,**

(i) $h(x) = h(y) \implies x \overset{*}{\leftrightarrow}_S y$,

(ii) S is almost confluent, and

(iii) L is an ACCL.

Definition. Define a map $\overline{(\ldots)}$ on Σ by

$$\overline{(a)} = \overline{a} \quad \overline{(b)} = \overline{b} \quad \overline{(\overline{a})} = a \quad \overline{(\overline{b})} = b$$

And extend it to an *anti-isomorphism* $\tilde{(\cdot)} : \Sigma^* \to \Sigma^*$:

$$x = a_1 a_2 \ldots a_k; \quad \tilde{x} = \overline{a_k}\,\overline{a_{k-1}} \ldots \overline{a_1}$$

Clearly $x\tilde{x} \overset{*}{\to}_S \lambda$ and $\tilde{x}x \overset{*}{\to}_S \lambda$.

Suppose (wrongly) that L is a CRCL, i.e., there exists a Church-Rosser Thue system T and strings $u_1, \ldots, u_n$ such that

$$L = [\lambda]_S = [u_1]_T \cup \ldots \cup [u_n]_T.$$

Important lemma. T refines S. (i.e., $x \overset{*}{\leftrightarrow}_T y \implies x \overset{*}{\leftrightarrow}_S y$).

Proof. Enough to show $[x]_S = [y]_S$ whenever $x \to_T y$. Suppose $x \to_T y$.

$$x\tilde{x} \to_T y\tilde{x} \quad \text{(clearly)},$$

$$x\tilde{x} \overset{*}{\to}_S \lambda \quad \text{(noted above)},$$

$$\therefore x\tilde{x} \in [\lambda]_S,$$

$$[\lambda]_S = [u_1]_T \cup \ldots \cup [u_n]_T,$$

$$\therefore (\exists i) \quad (x\tilde{x} \in [u_i]_T),$$

$$\text{but} \quad x \to_T y, \quad \text{so}$$

$$y\tilde{x} \in [u_i]_T$$

$$\therefore [y\tilde{x}]_S = [\lambda]_S;$$

$$\therefore [y\tilde{x}x]_S = [\lambda x]_S. \quad \text{Also,}$$

$$[y\tilde{x}x]_S = [y\lambda]_S.$$

$$\therefore [x]_S = [y]_S. \quad \blacksquare$$

Corollary. If $x \to_T y$ then $|y|_{\text{neg}} < |x|_{\text{neg}}$.

Proof. Let

$m_1 = |x|_{\text{pos}}$, $m_2 = |x|_{\text{neg}}$, $n_1 = |y|_{\text{pos}}$, $n_2 = |y|_{\text{neg}}$.

Since T refines S, $x \overset{*}{\leftrightarrow}_S y$, so

$$h(x) = h(y): \quad m_1 - m_2 = n_1 - n_2$$

Also, since T is length-reducing, $|x| - |y| > 0$, so

$$m_1 + m_2 > n_1 + n_2$$

$$m_1 - m_2 = n_1 - n_2$$

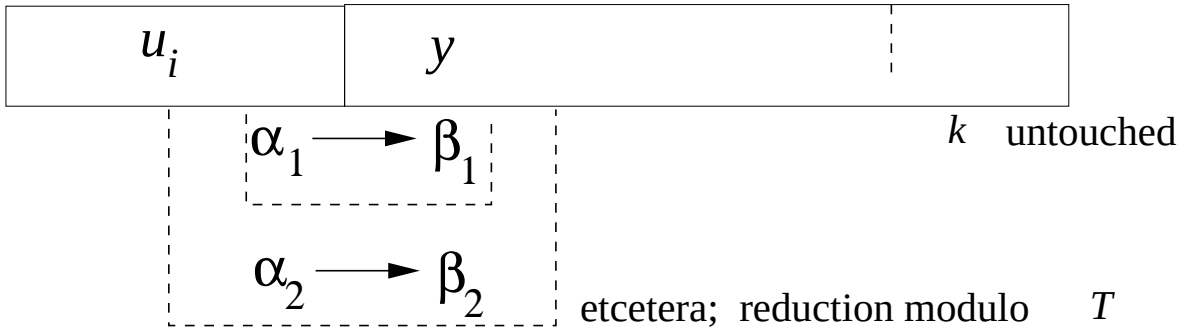
subtract: $2m_2 > 2n_2$, halve,

$$|y|_{\text{neg}} < |x|_{\text{neg}}. \quad \blacksquare$$

Lemma. Given a positive integer k, there exists a positive integer B so that if $y \in \text{POS}^*$

and $|y| = B$, and $1 \leq i \leq n$, and $u_i y \stackrel{*}{\rightarrow}_T z$ where z is irreducible (mod T), then y and z have the same length-k suffix.

(Rather easy proof omitted. See diagram.)



Completing proof that L cannot be

$$[u_1]_T \cup [u_2]_T \cup \ldots \cup [u_n]_T$$

Fix k so $2^k > n$, and B as above. Let $N = 2^k$. Choose any positive string x of length B. Let $x_1, \ldots, x_N$ be the complete list of strings which obtained by varying the last k letters of x, including x itself.

As usual, $x\tilde{x} \in [\lambda]_S$, so $[x\tilde{x}]_T = [u_i]_T$ for some i.

For $1 \leq j \leq N$,

$$[x\tilde{x}x_j]_T = [u_i x_j]_T \qquad (*)$$

$$\text{repeat}: \quad [x\tilde{x}x_j]_T = [u_i x_j]_T \qquad (*)$$

Since x and x_j are positive of the same length, $[\tilde{x}x_j]_S = [\lambda]_S$ (exercise), so for some ℓ, $1 \leq \ell \leq n$, $[\tilde{x}x_j]_T = [u_\ell]_T$, so

$$[x\tilde{x}x_j]_T = [xu_\ell]_T.$$

That is, there are at most n different classes on the left-hand side of $(*)$.

On the other hand, for $1 \leq j \leq N$ let z_j be the irreducible string in $[u_i x_j]_T$.

From a previous lemma, z_j and x_j share the same length-k suffix, so the strings z_j are all different and there are N classes on the right-hand side of $(*)$, and $n < 2^k = N$, which is impossible. ∎

Appeared: *Theoretical Computer Science* **589** *(2015)*, 141–146.