Some related work and an open (?) problem

Colm Ó Dúnlaing, T.C.D.

Notation: red for definition or first use, blue for explanations.

Alphabet Σ any finite set
Free monoid Σ^* semigroup with identity — under concatenation
Example Bitstrings: $\{0,1\}^*$, namely...
$\{\lambda, 0, 1, 00, 01, 10, \ldots\}$
Language L: any subset of any Σ^*

Examples
Even parity (bitstrings) $\{\lambda, 0, 00, 11, 000, \ldots\}$
Even length $\{\lambda, 00, 01, 10, 11, 0000, \ldots\}$
i.e., $\{x \in \{0,1\}^* : \quad |x| \equiv_2 0\}$
Palindromes $\{\lambda, 0, 1, 00, 11, 000, 010, \ldots\}$, i.e.,
$\{x \in \{0,1\}^* : \ x = x^R\}$.

Congruence $\equiv$ on Σ^* is an

- equivalence relation such that
- $\forall x, y$
  $x \equiv y \implies$
  $(\forall u, v) \quad uxv \equiv uyv.$

Congruence class $[x]_\equiv$ i.e. equivalence class
$\{y : \; x \equiv y\}$

Quotient monoid $M_\equiv$ (or $\Sigma^*/\equiv$)
$\{[x]_\equiv : \; x \in \Sigma^*\}$
associative operation $[x]_\equiv [y]_\equiv := [xy]_\equiv$

Given $L \subseteq \Sigma^*$, the linguistic congruence $\equiv_L$ is the coarsest congruence whose classes partition L. Specifically,
$x \equiv_L y \iff$ (def)
$\forall u, v \in \Sigma^* \quad (uxv \in L \Leftrightarrow uyv \in L)$

Examples (and exercises!)

$L =$ even-parity bitstrings: $x \equiv_L y$ iff they have the same parity

$L =$ even-length bitstrings: $x \equiv_L y$ iff $|x| \equiv_2 |y|$

$L =$ palindromes: $x \equiv_L y$ iff $x = y$.

Important definition

L is a Regular set if $M_{\equiv_L}$ is finite (i.e., $\equiv_L$ has finite index).

Exercise: the odd-parity bitstrings are a regular set; the odd-length bitstrings are a regular set; the palindromes aren't.

A Thue system T over Σ is any finite subset of $\Sigma^* \times \Sigma^*$ (any finite relation on Σ^*).

The next definitions are slightly different from the 1/7/09 talk, in response to some questions which arose during the talk

A Thue system can be considered as a finite description of a congruence, so Thue systems yield finitely generated congruences.

The Thue congruence $\overset{*}{\leftrightarrow}_T$ of a Thue system T is the coarsest congruence containing T.

Step-by-step: we write $x \leftrightarrow_T y$ to mean that there exists a rule (α, β) (i.e., a pair in T) and strings u and v such that either ($x = u\alpha v$ and $y = u\beta v$) or ($x = u\beta v$ and $y = u\alpha v$).

We use directed arrows to reflect *length*: given $x \leftrightarrow_T y$, write $x \longleftarrow_T y$, $x \longmapsfrom\!\!\!\!\longmapsto_T y$, $x \longmapsto_T y$, or $x \longrightarrow_T y$ according as $|x| < |y|$, $|x| = |y|$, $|x| \geq |y|$, or $|x| > |y|$. Thus

$$\leftrightarrow_T \quad = \quad \longleftarrow_T \vee \longmapsfrom\!\!\!\!\longmapsto_T \vee \longrightarrow_T .$$

and $\overset{*}{\leftrightarrow}_T$ is the reflexive and transitive closure of the relation $\leftrightarrow_T$.

The word problem for a Thue system T is to give an algorithm to decide for all x, y whether $x \overset{*}{\leftrightarrow}_T y$. In general it is recursively unsolvable: no such algorithm exists.

Example. One can present the free *group* on two generators a, b as a *monoid* on four generators $\Sigma = \{a, b, a^{-1}, b^{-1}\}$ with the following Thue system

$$\{(aa^{-1}, \lambda), (a^{-1}a, \lambda), (bb^{-1}, \lambda), (b^{-1}b, \lambda)\}.$$

These rules can be applied in different orders. For example the string $ab^{-1}baa^{-1}b$ can be reduced beginning in two ways, but, as illustrated, these diverging reductions can easily be reconciled, and in fact there is always only one irreducible outcome when we reduce a string (the freely reduced word).

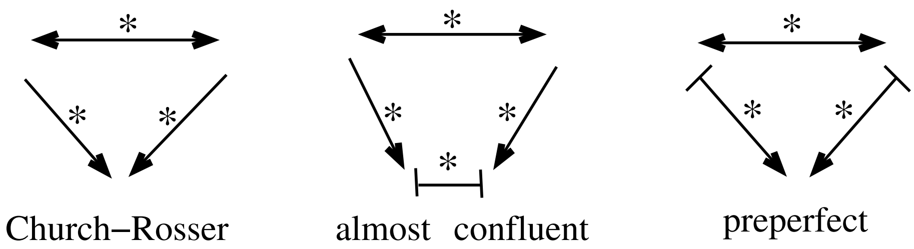
$$\begin{tikzcd}
& ab^{-1}baa^{-1}b \arrow[dl] \arrow[dr] & \\
aaa^{-1}b \arrow[dr, dotted, "*"] & & ab^{-1}bb \arrow[dl, dotted, "*"'] \\
& ab &
\end{tikzcd}$$

This Thue system is CR or Church-Rosser, meaning that whenever $x \overset{*}{\leftrightarrow}_T y$, there exists a

string z such that $x \xrightarrow{*}_T z$ and $y \xrightarrow{*}_T z$.

A Thue system is AC or almost confluent if whenever $x \xleftrightarrow{*}_T y$, there exist strings z_1 and z_2 such that $x \xrightarrow{*}_T z_1$ and $y \xrightarrow{*}_T z_2$ and $z_1 \overset{*}{\vdash\!\!\dashv}_T z_2$.

A Thue system is PP or preperfect if whenever $x \xleftrightarrow{*}_T y$, there exists a string z such that $x \overset{*}{\mapsto}_T z$ and $y \overset{*}{\mapsto}_T z$.



T is	word problem	is T?
CR	$O(n)$	$O(n^3)$
AC	PSPACE complete	in PSPACE
PP	ditto	undecidable

It is easy to decide whether a Thue system is CR, and recursively undecidable whether it is PP, mostly because $\rightarrow_T$ is noetherian but $\mapsto_T$ isn't.

A CRCL or Church-Rosser congruential language is a finite union of congruence classes from a Church-Rosser Thue system:

$$L = [x_1]_T \cup \ldots \cup [x_k]_T$$

where T is Church-Rosser.

An ACCL and a PPCL are defined similarly, where T is AC or PP respectively.

Small addition to notes: given a Thue system T, a string x is minimal if for all $y \in [x]_T$, $|y| \leq |x|$. Minimality is a property of the congruence rather than any system generating it, and is in general an undecidable property.

Suppose $\equiv$ is a congruence of *finite index* (closely related to a regular set).

Exercise: $\equiv$ is generated by a finite Church-Rosser Thue system if and only if every congruence class contains exactly one minimal string.

Exercise: $\equiv$ is always generated by a finite almost confluent Thue system (granted it has finite index).

Exercise: every regular set is an ACCL.

This brings us to the Open Question. (Or maybe open until recently). Is every regular set a CRCL?

Niemann and Waldmann (2001) showed that the language $L_2(a,b)$ (:= even-length strings in $\{a,b\}^*$) is a CRCL. Moreover...

An sCRCL or strongly CR congruential language L is (I'm not certain about this definition) one for which there exists a finite Church-Rosser Thue system T, of finite index — $|M_T| < \infty$ — such that $\overset{*}{\leftrightarrow}_T$ refines $\equiv_L$ (i.e.,

every congruence class of T is contained in one of $\equiv_L$). An sCRCL is a CRCL; it is (I believe) unknown whether the converse is true.

Niemann and Waldmann showed that L_2 is an sCRCL.

Exercise. The Thue system $T = \{(aaa, a), (bb, \lambda), (aba, a)\}$ is Church-Rosser and refines the congruence of L_2.

Schluter (2008) showed that $L_3 := \{x \in \{a, b\}^* : \ |x| \equiv_3 0\}$ is an sCRCL. Explicitly, the following Thue system is Church-Rosser and refines the linguistic congruence:

$aaaa \to a, \quad bbb \to \lambda, \quad abba \to a,$
$ababababa \to ababa, \quad abaaba \to aba,$
$ababaaaba \to ababaa,$ AND/OR
$abaaababa \to aababa,$
$abaaabaaabaaabaaabaaabaaaba \to aba$

The AND/OR refers to rules which are not both needed to ensure the Church-Rosser property. If both are included then $|M_T| \geq 252$;

else $|M_T| \geq 657$. (The exact cardinalities could easily be computed, but haven't been.)

Related work. Although the word problems for CR and AC systems have far different complexities, the membership problem for CRCLs and ACCLs are both $O(n)$. The hypothesis P$\neq$PSPACE is irrelevant.

A CRL (or Church-Rosser language) $L \subseteq \Sigma^*$ is any language for which there exists a Church-Rosser Thue system T, over an alphabet $\Gamma \supseteq \Sigma$ and strings $t_1, t_2, t_3 \in (\Gamma \backslash \Sigma)^*$, such that $t_1 L t_2 = [t_3]_T$. If $\Gamma = \Sigma$ then L is a CRCL, but the additional context t_1 and t_2 makes for a much broader class of languages.

Jurdziński and Loryś (2002, 2007) proved that the palindromes in $\{0, 1\}^*$ are not a CRL, using Kolmogorov complexity. They argue that palindromes cannot be recognised by deterministic shrinking automata, and the arguments can be interpreted as follows. Suppose T, t_1, t_2, t_3 are given. One can prove that for any sufficiently long incompressible

string w (in terms of Kolmogorov complexity) and any sufficiently high exponent d, the string $t_1(ww^R)^{2d+1}t_2$ ‘fools’ the Thue system for the following reasons. In every step of the form $x \to_T y$, the string y has less information than x, because of length reduction.

Middle block lemma: after sufficiently many reductions, one of the ‘blocks’ ww^R becomes ‘depleted,’ meaning that one cannot reconstruct w from it alone. This must be the middle block.

t_1	ww^R	ww^R	$\boxed{ww^R}$	ww^R	ww^R	t_2

Pumping lemma: at this point, one side (wlog the right) of the middle block can be ‘pumped,’ meaning that there exists a shorter string $t_1(ww^R)^d ww^R (ww^R)^e t_2$ producing a similar history, where $e < d$: this contradicts the middle block lemma.

Ó D and Schluter (2008) simplified and shortened the proof somewhat, and observed that the same proof worked for AC Thue

systems T, but there exists a *preperfect* Thue system T and strings t_1, t_2, t_3 such that $t_1 L t_2 = [t_3]_T$. This separates PPCLs from ACCLs but not ACCLs from CRCLs.

Additional remarks.

Exercise. If L is a regular set and $M_{\equiv_L}$ is a group, then L is a CRCL iff it is an sCRCL.

Exercise. The usual presentation $a^3 = b^2 = 1, a^2 b = ba$ of S_3 actually furnishes a Church-Rosser system without modification.

Exercise. Every finite monoid (correction: semigroup) is isomorphic to M_T for some Church-Rosser Thue system (correction: $\Sigma^+/\overset{*}{\leftrightarrow}_T$ — no λ). In other words, the problem is easy for isomorphism but not when the alphabet is fixed. This is opposite to the usual situation.

References.

Ronald V. Book and Friedrich Otto (1993). *String rewriting systems*. Springer.

G. Niemann and J. Waldmann (2001). Some regular languages that are Church-Rosser congruential. 5th DLT 2001 330-339 Vienna.

Colm Ó Dúnlaing and Natalie Schluter (2008). A shorter proof that palindromes are not a CRCL, with extensions to AC and PP Thue systems. Manuscript, submitted for publication.