strcmp between String Concatenation and String Repetition

Yi Tang

April 5, 2019

In this note we prove a property regarding strcmp, string concatenation, and string repetition. The problem is put forward in some private forum and the origin of this problem is unknown.

We start with introducing the basic definitions of strings and string operations.

Definition 1. A string a over alphabet Σ is a finite-length sequence over Σ, i.e., $a \in \Sigma^* := \bigcup_{n=0}^{\infty} \Sigma^n$. We fix some alphabet Σ (that is equipped with a total order) throughout this note and often omit it for simplicity. For strings a, b over the same alphabet, we have the following operations.

- The string concatenation ab is defined to be the concatenated sequence.
- For number $k \in \mathbb{N}$, the string repetition a^k is recursively defined to be $a^{k-1}a$ if $k > 0$ and the empty string ε if $k = 0$.
- The string length $|a|$ is defined to be the length of a as a sequence.
- For indices $i \le j \le |a| \in \mathbb{N}$, the substring $a_{i:j}$ is defined to be the corresponding subsequence (including i, excluding j, counting from 0); we also define $a_{:j} := a_{0:j}$ and $a_{i:} := a_{i:|a|}$.
- If the alphabet is totally ordered, then there exists a natural *lexicographic order* among the strings. The lexicographic order indicator $\mathsf{strcmp}(a, b)$ is defined to be -1 if $a < b$, 0 if $a = b$, and $+1$ if $a > b$.

We first show some handy properties of strcmp.

Lemma 2 ("Transitivity" of strcmp)**.** *For any strings a, b, c, if $\mathsf{strcmp}(a, b) = \mathsf{strcmp}(b, c)$, then*

$$\mathsf{strcmp}(a, b) = \mathsf{strcmp}(b, c) = \mathsf{strcmp}(a, c) \ .$$

Proof. Obvious according to the definition of strcmp. □

Lemma 3 (Decomposition of strcmp)**.** *For any strings a, b and index $k \le \min(|a|, |b|) \in \mathbb{N}$,*

$$\mathsf{strcmp}(a, b) = \mathsf{strcmp}(a_{:k}, b_{:k}) \mid\mid \mathsf{strcmp}(a_{k:}, b_{k:}) \ ,$$

where $x \mid\mid y$ denotes applying the quasi-logical operator "if $x \neq 0$ then x otherwise y".

Proof. Obvious according to the definition of lexicographic order. □

Corollary 4 (Padding of strcmp)**.** *For any strings a, b, c,*

$$\mathsf{strcmp}(a, b) = \mathsf{strcmp}(ca, cb) \ .$$

Furthermore, if $|a| = |b|$, then

$$\mathsf{strcmp}(a, b) = \mathsf{strcmp}(ac, bc) \ .$$

The corollary immediately follows the lemma, while we include a proof here to demonstrate how calculations can be carried out formally using the quasi-logical operator $\mid\mid$.

Proof. Using the lemma,

$$\mathsf{strcmp}(xa, xb) = \mathsf{strcmp}(x, x) \mathbin{||} \mathsf{strcmp}(a, b) = 0 \mathbin{||} \mathsf{strcmp}(a, b) = \mathsf{strcmp}(a, b) \ ,$$

and when $|a| = |b|$,

$$\mathsf{strcmp}(ay, by) = \mathsf{strcmp}(a, b) \mathbin{||} \mathsf{strcmp}(y, y) = \mathsf{strcmp}(a, b) \mathbin{||} 0 = \mathsf{strcmp}(a, b) \ . \qquad \square$$

Lemma 5 ("Conditional" strcmp)**.** *For any strings a, b and string functions f, g,*

$$\mathsf{strcmp}(a, b) \mathbin{||} \mathsf{strcmp}(f(a), g(a)) = \mathsf{strcmp}(a, b) \mathbin{||} \mathsf{strcmp}(f(b), g(b)) \ .$$

Proof. If $\mathsf{strcmp}(a, b) \neq 0$, then both the LHS and the RHS equal $\mathsf{strcmp}(a, b)$. Otherwise, $\mathsf{strcmp}(a, b) = 0$ implies $a = b$, and thus the LHS and the RHS are equal as well. $\square$

Now we present the property regarding strcmp, string concatenation, and string repetition, which is the protagonist of this note.

Theorem 6 (strcmp between string concatenation and string repetition)**.** *For any strings a, b,*

$$\mathsf{strcmp}(ab, ba) = \mathsf{strcmp}(a^{|b|}, b^{|a|}) \ .$$

Proof. Without loss of generality suppose $|a| \leq |b|$. We prove by induction on $|a|+|b|$. Note that the equality is obvious when $a = \varepsilon$ or $b = \varepsilon$, i.e. $|a| + |b| \leq 1$. Assume the equality holds for all a, b such that $|a| + |b| \leq k$ ($k \geq 1 \in \mathbb{N}$). Then for $|a| + |b| = k + 1$, by decomposition of and "conditional" strcmp,

$$\begin{aligned}
\mathsf{strcmp}(ab, ba) &= \mathsf{strcmp}(a, b_{:|a|}) \mathbin{||} \mathsf{strcmp}(b, b_{|a|:}a) \\
&:= \mathsf{strcmp}(a, b_1) \mathbin{||} \mathsf{strcmp}(b_1 b_2, b_2 a) \\
&= \mathsf{strcmp}(a, b_1) \mathbin{||} \mathsf{strcmp}(a b_2, b_2 a) \ ; \\
\mathsf{strcmp}(a^{|b|}, b^{|a|}) &= \mathsf{strcmp}(a^{|b_1|+|b_2|}, (b_1 b_2)^{|a|}) \\
&= \mathsf{strcmp}(a, b_1) \mathbin{||} \mathsf{strcmp}(a^{|b_1|+|b_2|-1}, (b_2 b_1)^{|a|-1} b_2) \\
&= \mathsf{strcmp}(a, b_1) \mathbin{||} \mathsf{strcmp}(a^{|b_2|+|a|-1}, (b_2 a)^{|a|-1} b_2) \ .
\end{aligned}$$

If $|a| = 1$ then $\mathsf{strcmp}(a^{|b_2|+|a|-1}, (b_2 a)^{|a|-1} b_2) = \mathsf{strcmp}(a^{|b_2|}, b_2^{|a|})$, which equals $\mathsf{strcmp}(ab_2, b_2 a)$ by inductive assumption. When $|a| > 1$, note that for all $j \leq i < |a| - 1 \in \mathbb{N}$, by padding of strcmp,

$$\begin{aligned}
\mathsf{strcmp}(ab_2, b_2 a) = \mathsf{strcmp}(&a^i b_2^{i-j} (ab_2) b_2^j (b_2 a)^{|a|-i-2} b_2, \\
&a^i b_2^{i-j} (b_2 a) b_2^j (b_2 a)^{|a|-i-2} b_2) \ ,
\end{aligned}$$

which gives a transitioning from $a^{|a|-1} b_2^{|a|}$ ($i = j = |a| - 2$) to $(b_2 a)^{|a|-1} b_2$ ($i = j = 0$). Therefore by "transitivity" of strcmp,

$$\mathsf{strcmp}(ab_2, b_2 a) = \mathsf{strcmp}(a^{|a|-1} b_2^{|a|}, (b_2 a)^{|a|-1} b_2) \ .$$

Moreover, by inductive assumption,

$$\begin{aligned}
\mathsf{strcmp}(ab_2, b_2 a) &= \mathsf{strcmp}(a^{|b_2|}, b_2^{|a|}) \\
&= \mathsf{strcmp}(a^{|a|-1} a^{|b_2|}, a^{|a|-1} b_2^{|a|}) \\
&= \mathsf{strcmp}(a^{|b_2|+|a|-1}, a^{|a|-1} b_2^{|a|}) \ .
\end{aligned}$$

Then again by "transitivity,"

$$\mathsf{strcmp}(ab_2, b_2 a) = \mathsf{strcmp}(a^{|b_2|+|a|-1}, (b_2 a)^{|a|-1} b_2) \ .$$

Hence in both cases ($|a| = 1$ and $|a| > 1$), $\mathsf{strcmp}(ab_2, b_2 a) = \mathsf{strcmp}(a^{|b_2|+|a|-1}, (b_2 a)^{|a|-1} b_2)$ and thus $\mathsf{strcmp}(ab, ba) = \mathsf{strcmp}(a^{|b|}, b^{|a|})$, which completes the induction. $\square$