Question 1: Playing around with Strings

Show every step of your calculations for the following questions. Please always complete your calculations; for example, in your answer write 3125 instead of 5^5.

(a) Which integer is associated with the string “zork” on the alphabet $A = \{a, o, r, k, z\}$?

The letters in “zork” are symbols number 5, 2, 3, and 4 in the given alphabet $A$. Since $A$ contains 5 letters, we have $n = 5$. Then the integer $x$ we are looking for can be computed as:

$$x = 5 \cdot 5^3 + 2 \cdot 5^2 + 3 \cdot 5^1 + 4 \cdot 5^0 = 625 + 50 + 15 + 4 = 694$$

(b) What is the string on the alphabet $B = \{a, b, c\}$ that is associated with the integer 134?

$B$ has 3 symbols, and therefore, $n = 3$. We use our division algorithm with $Q^+$ and $R^+$:

$$
\begin{align*}
134/3 &= 44 \ R \ 2 \\
44/3 &= 14 \ R \ 2 \\
14/3 &= 4 \ R \ 2 \\
4/3 &= 1 \ R \ 1 \\
1/3 &= 0 \ R \ 1 
\end{align*}
$$

The string thus consists of symbols number 1, 1, 2, 2, and 2, which in the given alphabet $B$ is “aabbbaa”

(c) Compute $\text{UPCHANGE}_{3,5}(100)$.

First, we need to compute the base 3 extension of 100:

$$
\begin{align*}
100/3 &= 33 \ R \ 1 \\
33/3 &= 11 \ R \ 0 \\
11/3 &= 3 \ R \ 2 \\
3/3 &= 1 \ R \ 0 \\
1/3 &= 0 \ R \ 1
\end{align*}
$$

The base 3 extension of 100 is 3201010.
\[ 3/3 = 0 \text{ R } 3 \]

This gives us the string \( s_3s_1s_3s_1 \).

We now have to find the integer \( x \) whose base 5 extension is \( s_3s_1s_3s_1 \):

\[ x = 3 \cdot 5^3 + 1 \cdot 5^2 + 3 \cdot 5^1 + 1 \cdot 5^0 = 375 + 25 + 15 + 1 = 416 \]

(d) Compute \( \text{DOWNCHANGE}_{7,9}(201) \).

Base 9 extension of 201:

\[
\begin{align*}
201/9 & = 22 \text{ R } 3 \\
22/9 & = 2 \text{ R } 4 \\
2/9 & = 0 \text{ R } 2
\end{align*}
\]

So we get \( s_2s_4s_3 \). In base 7 extension, this is:

\[ x = 2 \cdot 7^2 + 4 \cdot 7^1 + 3 \cdot 7^0 = 98 + 28 + 3 = 129 \]

(e) Compute \( \text{LTEND}_4(37) \).

(f) \( \text{LTEND} \) returns the leftmost symbol of a given string. So let us first represent 37 in base 4:

\[
\begin{align*}
37/4 & = 9 \text{ R } 1 \\
9/4 & = 2 \text{ R } 1 \\
2/4 & = 0 \text{ R } 2
\end{align*}
\]

The result is \( s_2s_1s_1 \). Thus, the leftmost symbol is \( s_2 \) and the solution is 2.

(g) Given the alphabet \( C = \{ x, y, z \} \), what is the least integer whose associated string on \( C \) has a length of four?

The string of length four with the least associated integer is the one that only contains the first (least valued) symbol in the alphabet. Here, this is the string \( xxxx \).

Since we have \( n = 3 \), the integer \( x \) associated with that string is:

\[ x = 1 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3^1 + 1 \cdot 3^0 = 27 + 9 + 3 + 1 = 40 \]
Question 2: Ordering with $L_2$

Write an $L_2$ program (you can use macros) that computes the following function $f$:

$$f(x, y) = \begin{cases} s_1, & \text{if } x < y \\ s_2, & \text{otherwise} \end{cases}$$

Here, $x < y$ means that the string $x$ precedes the string $y$ in alphabetical order. For example, it is true that $s_1s_2s_2s_1 < s_2s_1$, $s_1s_1s_1 < s_1s_1s_2$, and $s_2s_2 < s_2s_2s_2$.

The easiest way to do this (in my opinion) is to compare the leftmost symbols of the string, and if they differ, we already know the answer. Otherwise, we remove the leftmost symbols from both strings and repeat this procedure. At some point, one of three things must happen:

1. The symbols differ and we can decide as above;
2. One string runs out of symbols while the other still has at least one left. In that case, the one running out of symbols is the smaller one;
3. Both strings run out of symbols at the same time, and thus they are identical, i.e., $x = y$.

Here is an example program:

[A1] IF $X_1 \neq 0$ GOTO D1
    IF $X_2 \neq 0$ GOTO C1
[B1] $Y \leftarrow s_2Y$                     // Strings are identical, so return $s_2$
    GOTO E
[C1] $Y \leftarrow s_1Y$                     // Strings are identical except that $X_2$ contains additional symbols to the right, so $X_1 < X_2$, and we return $s_1$
    GOTO E
[D1] IF $X_2 \neq 0$ GOTO A2
    GOTOB1                      // This time, $X_1$ contains additional symbols, so return $s_2$
[A2] $Z_1 \leftarrow \text{LTEND}_2(X_1)$    // Get the leftmost symbol of $X_1$
    $X_1 \leftarrow \text{LTRUNC}_2(X_1)$    // Remove that symbol from $X_1$
    $Z_2 \leftarrow \text{LTEND}_2(X_2)$    // Get the leftmost symbol of $X_2$
    $X_2 \leftarrow \text{LTRUNC}_2(X_2)$    // Remove that symbol from $X_2$
    IF $Z_1$ ENDS $s_1$ GOTO A3
    IF $Z_2$ ENDS $s_1$ GOTO B1          // $X_1 > X_2$, so return $s_2$
    GOTO A1                      // Strings are identical so far, so check next pair of symbols
[A3] IF $Z_2$ ENDS $s_1$ GOTO A1
    GOTO C1                      // $X_1 < X_2$, so return $s_1$
Question 3 (Bonus Question): Ordering with Post-Turing

a) Write a Post-Turing program (you can use macros) using the alphabet $A = \{s_1, s_2\}$ that computes the same function $f$ from Question 2.

We can use the same algorithm as in Question 2. Since the computation does not have to be strict, we can use a marker symbols $M$ and $N$ to overwrite those symbols on the first and second input string, respectively, that we have already compared.

Example program:

```
[A1] RIGHT
    IF s_1 GOTO B1
    IF s_2 GOTO C1
    PRINT M // Remaining string X_1 is empty

[A2] RIGHT
    IF B GOTO D1 // If there are two consecutive Bs, X_2 is empty, so output s_2
    IF N GOTO A2

[A3] PRINT B // There is an additional symbol in X_2, so output s_1
    RIGHT // Remove all symbols to the right…
    IF s_1 GOTO A3
    IF s_2 GOTO A3

[A4] PRINT B // … then remove all symbols to the left…
    LEFT
    IF M GOTO A5
    GOTO A4

[A5] PRINT B // … then remove all remaining M markers …
    LEFT
    IF M GOTO A5
    PRINT s_1 // … and finally output s_1
    GOTO E

[B1] PRINT M

[B2] RIGHT // Find next B to the right
    IF B GOTO B3
    GOTO B2

[B3] RIGHT // Find next symbol s_1 or s_2 to the right
    IF N GOTO B3
    IF B GOTO D1 // X_1 > X_2, so output s_2
    IF s_2 GOTO A3 // X_1 < X_2, so output s_1

[B4] PRINT N // X_1 = X_2 so far, compare next pair of symbols

[B5] LEFT // Find next M to the left…
    IF M GOTO A1 // … and then start next comparison
    GOTO B5

[C1] PRINT M
[C2] RIGHT // Find next B to the right
```
IF B GOTO C3
GOTO C2
[C3]  RIGHT  // Find next symbol s1 or s2 to the right
IF N GOTO C3
IF B GOTO  // X1 > X2, so output s2
IF s1 GOTO  // X1 > X2, so output s2
GOTO B4  // X1 = X2 so far, compare next pair of symbols
[D1]  PRINT B  // output s2
       RIGHT  // Remove all symbols to the right…
       IF s1 GOTO D1
       IF s2 GOTO D1
[D2]  PRINT B  // … then remove all symbols to the left…
       LEFT
       IF M GOTO D3
       GOTO D2
[D3]  PRINT B  // … then remove all remaining M markers …
       LEFT
       IF M GOTO D3
       PRINT s2  // … and finally output s2

b) Write down the list of successive tape configurations that your program generates during the computation of f(s1s2s1, s1s2s2s1).