Numerical Representation of Strings

4.) \( RTEND_n(w) = h(0, n, w) \)

Remember:
\[ i_m = h(m, n, x), \quad m = 0, \ldots, k. \]

\( RTEND_n \) gives the rightmost symbol of a given word.

We know that \( h \) is primitive recursive, so \( RTEND_n \) is also primitive recursive.

Numerical Representation of Strings

5.) \( LTEND_n(w) = h(|w| - 1, n, w) \)

Corresponding to \( RTEND_n \), \( LTEND_n \) gives the leftmost symbol of a given word.

Numerical Representation of Strings

6.) \( RTRUNC_n(w) = g(1, n, w) \)

Remember:
\[
\begin{align*}
    u_0 &= i_k n^k + i_{k-1} n^{k-1} + \ldots + i_1 n + i_0 \\
    u_1 &= i_k n^{k-1} + i_{k-1} n^{k-2} + \ldots + i_1
\end{align*}
\]

\( RTRUNC_n \) gives the result of removing the rightmost symbol from a given nonempty string.

When we can omit the reference to the base \( n \), we often write \( w^- \) for \( RTRUNC_n(w) \).

Note that \( 0^- = 0 \).

Numerical Representation of Strings

7.) \( LTRUNC_n(w) = w^- (LTEND_n(w) \cdot n^{|w| - 1}) \)

\( LTRUNC_n \) gives the result of removing the leftmost symbol from a given nonempty string.

Example: Alphabet \( A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, X\} \):
\[ LTRUNC_{10}(3478) = 478. \]

We will use these newly introduced primitive recursive functions to prove the computability of a pair of functions that can be used in changing base.

Let \( 1 \leq n < l \).

Let \( A \subset \bar{A} \), where \( A \) is an alphabet of \( n \) symbols and \( \bar{A} \) is an alphabet of \( l \) symbols.

So whenever a string belongs to \( A^* \), it also belongs to \( \bar{A}^* \).

For any \( x \in \mathbb{N} \), let \( w \) be the word in \( A^* \) that represents \( x \) in base \( n \).

Then we write \( UPCHANGE_n(x) \) for the number which \( w \) represents in base \( l \).

Examples:
\[ UPCHANGE_{2,5}(5) = 13 \]

The representation of 5 in base 2 is \( s_2s_1 \). In base 6, \( s_2s_1 \) represents the number 13.

\[ UPCHANGE_{5,2}(3) = 31 \]

The representation of 3 in base 1 is \( s_1s_1s_1 \). In base 5, \( s_1s_1s_1 \) represents the number 31.
Numerical Representation of Strings
Correspondingly, we define the following:
For \( x \in \mathbb{N} \), let \( w \) be the string in \( \hat{A}^* \) which represents \( x \) in base \( l \).
Let \( w' \) be obtained from \( w \) by crossing out all of the symbols that belong to \( \hat{A} - A \), so \( w' \in A^* \).
We write \( \text{DOWNCHANGE}_{n,l}(x) \) for the number which \( w' \) represents in base \( n \).
Example:
\[ \text{DOWNCHANGE}_{2,6}(109) = 5 \]
The representation of 109 in base 6 is \( s_2s_6s_1 \). We cross out the \( s_6 \) and get \( s_2s_1 \). In base 2, \( s_2s_1 \) represents the number 5.

Numerical Representation of Strings
UPCHANGE_{n,l} and DOWNCHANGE_{n,l} are actually primitive recursive functions, but now we will just show that they are computable.
This program computes \( \text{UPCHANGE}_{n,l} \):

\[
[A] \quad \text{IF } X = 0 \text{ GOTO } E
\]
\[ Z \leftarrow \text{LTEND}_{n}(X) \] // \( Z \) receives leftmost symbol
\[ X \leftarrow \text{LTRUNC}_{n}(X) \] // removes this symbol from \( x \)
\[ Y \leftarrow l \cdot Y + Z \] // add to output
\[ \text{GOTO } A \]

Numerical Representation of Strings
Let us look at the computation of \( \text{UPCHANGE}_{2,10}(12) \) iteration by iteration (string \( s_2s_1s_2 \)):

\[ [A] \quad \text{IF } X = 0 \text{ GOTO } E \]
\[ Z \leftarrow \text{LTEND}_{10}(X) \] 0. \( Z = 0, Y = 0, X = 12 \)
\[ X \leftarrow \text{LTRUNC}_{10}(X) \] 1. \( Z = 2, Y = 2, X = 4 \)
\[ Y \leftarrow l \cdot Y + Z \] 2. \( Z = 1, Y = 21, X = 2 \)
\[ \text{GOTO } A \] 3. \( Z = 2, Y = 212, X = 0 \)
Result: 212