ISIS-1106 Lenguajes y máquinas
Project 2
Fecha: November 5, 2021

The purpose of this project is to use GOLD to define finite state transducers to code and decode inputs.

Task 1. The task of this project is to define the coder and decoders described below:

1. The coder reads strings of the form $\sigma\alpha : \omega\$$ where:

    - $\sigma \in ('a'..'z')$
    - $\alpha \in ('a'..'z')$
    - $\omega = \sigma_1 \ldots \sigma_n$, with $\sigma_i \in ('a'..'z')$

    The coder should output: $\sigma\alpha\beta d$, where $\beta = \rho_1 \ldots \rho_n$.

    - Each ρ_i is coded as follows:
        - if $\sigma_i = \sigma$ then $\rho_i = \alpha$
        - if $\sigma_i = \alpha$ then $\rho_i = \sigma$
        - if $\sigma_i \notin \{\sigma, \alpha\}$
            * if $\sigma_i = \sigma_{i-1}$ then $\rho_i = \#$
            * if $\sigma_i \neq \sigma_{i-1}$ and $(i\%3) = 1$ then ρ_i is the uppercase of the letter after σ_i (i.e., if $\sigma_i = a$, $\rho_i = B$; if $\sigma_i = b$, $\rho_i = C$, $\ldots$ if $\sigma_i = z$, $\rho_i = A$)
            * if $\sigma_i \neq \sigma_{i-1}$ and $(i\%3) = 2$ then ρ_i is the upper case of σ_i
        - if none of the conditions apply, then $\rho_i = \sigma_i$
    - d is the number of replacements (σ for α and α for σ) that were made modulo 5

2. The decoder should read coded strings, verify that the string was coded correctly and decode the string.

We include a coder-decoder gold project. You may use it as a starting point.

Below, you will find some examples for the coder.

```
----------------------------
Input string: (empty string to end)
ac:babcaannnnnn$
The string was accepted. Output: ac:CcbaccO####4
----------------------------
Input string: (empty string to end)
ft:aaaaaaa$
The string was accepted. Output: ft:B######0
----------------------------
Input string: (empty string to end)
ma:amamamjkslell$
The string was accepted. Output: ma:mamamaKKsMEl#1
----------------------------
Input string: (empty string to end)
tt:atatataaat$
The string was accepted. Output: tt:BtatAtB##t4
----------------------------
Input string: (empty string to end)
az:aaaaabbbababz$
The string was accepted. Output: az:zzzzzb##zCzba3
----------------------------
Input string: (empty string to end)
xd:amaxmadcatisveeerysweet$
The string was accepted. Output: xd:BMadMaxCaUIsWE##RyTWe#T2
----------------------------
Input string: (empty string to end)
ch:acatishome$
The string was accepted. Output: ch:BhaUIscOmF2
----------------------------
```