LÍNGUA NATURAL 2022/2023

Mini-Project Nº 1 (MP1)

Should be done:	☐ individually	☒ **in group**
Submission:	☐ theoretical class	☒ **Fenix submission**
Submission deadline:	**till 23:59, October 10th**	

OBJECTIVES

Learn to work with transducers, using them to solve a problem.

STATEMENT

The so-called phonetic algorithms aim to associate a single representation (key) with words that sound similar. The *Soundex* (Odell and Russell, 1922) and the *Metaphone* (Lawrence Philips, 1990) are examples of this type of algorithm. For example:

- JRFSK — represents Jurafsky, Jarofsky, Jarovsky e Jarovski
- NN — represents Nuno, Nunu, Nonu, Nono

Can you recognize this text?

> T B, OR NT T B,
> 0T IS 0 KSKSN
> B WLM SHKSPR

In this work, we are going to define a variation (https://en.wikipedia.org/wiki/Metaphone) of the original Metaphone. This code uses the 16 symbols 0BFHJKLMNPRSTWXY. The '0' (zero) represents "th" (as an ASCII approximation of Θ), 'X' represents "sh" or "ch", and the others represent their usual English pronunciations. The vowels AEIOU are also used, but only at the beginning of the word.

1. Develop the following transducers:
   a. the transducer **step1** performs the following transformations:
      - drop duplicated adjacent letters, except for 'C' (ex: *ERRORS* → *ERORS*, *ACCORDING* → *ACCORDING*, *RUSSIAN* → *RUSIAN*).

   b. the transducer **step2** performs the following transformations:
      - if the word begins with 'KN', 'GN', 'PN', 'AE', or 'WR', drop the first letter (ex: *KNEE* → *NEE*, *GNOME* → *NOME*, *WRAPPERS* → *RAPPERS*);
      - if the word ends with 'MB' drop the 'B'. (ex: *BREADCRUMB* → *BREADCRUM*).

   c. the transducer **step3** performs the following transformations:
      - 'C' transforms to 'K' if in-between 'S' and 'H' ('-SCH-') (ex: *SCHOOL* → *SKHOOL*);
      - 'C' transforms to 'X' if followed by 'H' (and it is not part of '-SCH-') (ex: *ACHIEVER* → *AXHIEVER*);
      - 'C' transforms to 'X' if followed by 'IA' (ex: *PRONUNCIATION* → *PRONUNXIATION*);
      - 'C' transforms to 'S' if followed by 'I' (and it is not part of '-CIA-') (ex: *VICIOUS* → *VISIOUS*);
      - 'C' transforms to 'S' if followed by 'E' (ex: *ABSENCE* → *ABSENSE*);
      - 'C' transforms to 'S' if followed by 'Y' (ex: *CYBERNETICIAN* → *SYBERNETIXIAN*);
      - Otherwise, 'C' transforms to 'K' (ex: *CULTURE* → *KULTURE*).

   d. the transducer **step4** performs the following transformations:
      - 'D' transforms to 'J' if followed by 'GE' (ex: *PLEDGES* → *PLEJGES*),
      - 'D' transforms to 'J' if followed by 'GY' (ex: *FUDGY* → *FUJGY*),
      - 'D' transforms to 'J' if followed by 'GI' (ex: *BUDGIES* → *BUJGIES*),
      - Otherwise, 'D' transforms to 'T' (ex: *ABDUCED* → *ABTUCET*, *AID* → *AIT*).

   e. the transducer **step5** performs the following transformations:
      - Drop 'G' if followed by 'H' and 'H' is not at the end (ex: *FIGHT* → *FIHT*);
      - if the word ends with 'GN' drop the 'N'. (ex: *FOREIGN* → *FOREIN*);
      - if the word ends with 'GNED' drop the 'G'. (ex: *SIGNED* → *SINED*).

   f. the transducer step6 performs the following transformations:

f. the transducer **step6** performs the following transformations:
   - Drop 'H' if after a vowel and not before a vowel (ex: *FIHT* → *FIT*, *MAHARAJAH* → *MAHARAJA*);
   - 'CK' transforms to 'K' (ex: *LUCK* → *LUK*);
   - 'PH' transforms to 'F' (ex: *PHOTO* → *FOTO*);
   - 'Q' transforms to 'K' (ex: *QUITE* → *KUITE*);
   - 'S' transforms to 'X' if followed by 'H' (ex: *SHOULD* → *XHOULD*);
   - 'S' transforms to 'X' if followed by 'IO' (ex: *COMISIONER* → *COMIXIONER*);
   - 'S' transforms to 'X' if followed by 'IA' (ex: *RUSIA* → *RUXIA*).

g. the transducer **step7** performs the following transformations:
   - 'T' transforms to 'X' if followed by 'IA' (ex: *SUBSTANTIAL* → *SUBSTANXIAL*);
   - 'T' transforms to 'X' if followed by 'IO' (ex: *CALCULATION* → *CALCULAXION*);
   - 'TH' transforms to '0' (zero). (ex: *THE* → *0E*);
   - Drop 'T' if followed by 'CH' (ex: *MATCH* → *MACH*);
   - 'V' transforms to 'F' (ex: *HAVE* → *HAFE*);
   - 'WH' transforms to 'W' if at the beginning (ex: *WHAT* → *WAT*).

h. the transducer **step8** performs the following transformations:
   - 'X' transforms to 'S' if at the beginning. (ex: *XENON* → *SENON*);
   - Otherwise, 'X' transforms to 'KS' (ex: *SEX* → *SEKS*);
   - Drop 'W' if not followed by a vowel (ex: *LAWN* → *LAN*);
   - Drop 'Y' if not followed by a vowel (ex: *BY* → *B*, *KEYBOARD* → *KEBOARD*);
   - 'Z' transforms to 'S' (ex: *SIZE* → *SISE*).

i. the transducer **step9** performs the following transformations:
   - Drop all vowels unless it is the beginning (ex: *USE* → *US*, *KEBOARD* → *KBRD*).

2. To implement a simplification of the Metaphone phonetic algorithm, define the transducer **metaphoneLN**, using the previous smaller transducers that must be applied using the order of presentation.

3. Test the transducer **metaphoneLN** with the first and last names of each member of the group. The filenames of all of the files used in your tests should start with the prefix **"t-"** followed by the corresponding IST student number (5 or 6 digits). For example, t-12345-std1-in.str, t-12345-std1-in.txt, t-12345-std1-in.fsm, t-12345-std1-in.pdf, t-12345-std1-out.fsm, t-12345-std1-out.pdf.

4. Generate the transducer **invertMetaphoneLN**, the inversion of **metaphoneLN**, and test it. The report must contain a brief analysis of its usability.

Assume that:

- The "syms.txt" file contains the symbols to be manipulated by the transducers and cannot be changed;
- All transducers process one word at a time (and not a sequence of words);
- The first 9 transducers cannot make conversions beyond what is defined in their specification;
- You can use other transducers not mentioned in the statement;
- Within each step, the order in which the rules are applied is irrelevant;
- The input tape contains at least one character;
- The input tape never contains sequences of 3, or greater, similar letters (ex: 'AAA', 'AAAAA');
- The input tape can contain a sequence of letters that do not correspond to a valid English word (ex: 'BAABBABAA');
- The "output tape" must always contain a single output (must be deterministic);
- Input and output must be uppercase;
- The names of the transducers must be exactly: **step1**, **step2**, **step3**, **step4**, **step5**, **step6**, **step7**, **step8**, **step9**, **metaphoneLN**, and **invertMetaphoneLN**.

Software

To test the proposed solution use, in a Linux environment, the tools:

- "OpenFST" from Google (http://www.openfst.org/twiki/bin/view/FST/FstDownload).
- "Graphviz" (http://www.graphviz.org/);

Submission

Submit in Fenix, project *MP1*, a zip file with, and only with:

- a shell script [the name has to be "run.sh"] with **all** the commands used to generate all transducers, either in binary and in graphical format (PDF, PS, or PNG) from the ".txt" files;
- a folder "friendly" containing all the text files to be used with "compact2fst.py" to generate sources (extension

".txt");

- a folder "sources" containing all the text files used to define the transducers (extension ".txt");
- a folder "tests" with all the source test files (extensions ".txt"). It may also include files with extensions ".str" if you decide to use *word2fst.py*. This script receives as input "*.str" and generates as output "*.txt");
- a folder "compiled" containing all the compiled versions of all the transducers used, including the tests (extension ".fst");
- a folder "images" containing the graphical versions of all the transducers, including the tests (extension ".pdf", ".ps" or ".png");
- a short report, with the following requirements:
  - the filename has to be "report.txt" or "report.pdf";
  - must not exceed one page;
  - must identify the members of the group with an estimate of each element's contribution to the work. For example, Peter: 60%, John: 40%, along with a short justification;
  - must contain a brief description of your options;
  - must contain comments on the viability of your solution (is usable?);
  - must contain a brief analysis of the usability of the "inverted" metaphoneLN transducer, with a maximum of half a page.

You can make several submissions: a new submission replaces the previous one.

Attention:

- developed transducers must have exactly the same names as above;
- the 4 folders "sources", "tests", "compiled" and "images" should not contain sub-folders.

Evaluation criteria

The following criteria will be taken into account in the assessment (maximum = 20 points):

1. Correct operation of each **step1-9** transducer (1,5 points each);
2. Correct operation of the **metaphoneLN** and **invertMetaphoneLN** transducer (1,5 points);
3. Run.sh operating correctly (3 points);
4. Submission of the graphic versions of all transducers, as well as the examples, in their different forms, that is, before and after being fed as input to the **metaphoneLN** transducers (1 point);
5. Quality of the report [in Portuguese or English] including spelling and syntactic correction (1 point);

Non-compliance with any rule implies a minimum discount of 4 points (in 20 points).

During the evaluation of the correct operation of any transducer, the evaluation does not take into account the origin of the errors (e.g., when testing a transducer B, every time the expected output is not obtained, an error is taken into account, even when the origin of the error is the malfunction of another transducer used to generate B). So, malfunctions on the first 9 transducers may have an impact on the evaluation of the **metaphoneLN** transducer.

"Academic Integrity" in Língua Natural

In this course, each student is expected to subscribe to the highest standards of academic honesty. This means that every idea that is not the student's must be explicitly accredited to the respective author. Failure to do so constitutes plagiarism.

Plagiarism includes using ideas, code, or sets of solutions from other students or individuals, or any sources other than the course texts, without crediting those sources. Students are encouraged to discuss the problems with other students and should mention this discussion when they submit their results. This mention will NOT influence the grade. Students should not, under any circumstances, show to their classmates, even temporarily, their solutions to the quizzes or projects subject to evaluation. They should not even throw away drafts of the solutions without destroying them first, nor leave the developed code on shared-use computers.

Academic dishonesty also includes copying in exams. In this discipline, these should be taken without consulting any text or other classmates. Receiving or giving help during these exams is an act of academic dishonesty. Situations that could give rise to suspicions of dishonesty (opening backpacks to get paper, looking around instead of concentrating on the exam paper, etc.) should be avoided.

In this course, academic dishonesty is considered fraud, with all the legal consequences. Any fraud will have the immediate consequence of failing all students involved (including those who enabled it to occur). Any suspicion of academic dishonesty will be reported to the higher bodies of the school for disciplinary action. This may result in failure of the subject, failure of the year, temporary or permanent suspension from IST, or even from the University of Lisbon.

Copyright in Moodle and Fenix