Processamento de Linguagens e Compiladores (3º Ano LCC)
Project 1
Project Report

Bruno Dias da Gião
A96544

Maria Filipa Rodrigues
A97536

November 2, 2022

Resumo

Um ficheiro do tipo *Comma-Separated Values* é um formato de extrema importância, isto devido ao facto de ser texto plano, no entanto, as suas limitações são limitadas, motivando a escrita de um programa em Python que, usando expressões regulares, converte qualquer ficheiro deste tipo para um ficheiro do tipo *JavaScript Object Notation*, que, devido à sua legibilidade e capacidade de ser *parsed* diretamente para um objecto de JavaScript, tem praticalidade mais acrescida. Isto, claro, representa uma conversão extremamente trivial e, devido à natureza de CSV, pode até permitir a conversão de qualquer ficheiro variante de Excel Binary File Format para um ficheiro JSON.

Abstract

A Comma-Separated Values file is an extremely important file type, this is due to it being plain text, however it's applications are limited, motivating us to writing a program in Python that, using regular expressions, converts any file with this file format into a JavaScript Object Notation file, which has more practical uses for both it's readability and being parsed into JavaScript Objects. This, of course, has an extremely trivial conversion and due to CSV's nature, might even allow for the conversion of any XLS file variation into JSON.

Contents

Chapter 1

Report

1.1 Introduction

1.1.1 CSV to JSON conversion

Introduction to the Report

This report is structured by the literate component itself 1 and by the code component which shall contain all the code used for this project 2.
Within the literate component we introduce the project with a historical background of the file formats being studied, why we believe this project is important, the original premise of the exercise and what can be done to expand it in a productive way that complements what is lectured in this class.
After this introduction, we move on towards implementations, design decisions and the finer technical and theoretical aspects of the project in the methodology section 1.2. Here we go into detail on regular expressions, the re module, how the properties of csv were used to manipulate the file via regex and how the json file was created. In this very section there is also a segment dedicated to exemplifying the tests used to verify the good behaviour of the program.
The last section of the this chapter is the conclusion 1.3 where a brief summary of the results are gathered and insight is given on what could've been done and how this project can be expanded on.
Finally, we have the code component of the report which will contain all the code used during this project.
As standard for a report, the last pages are dedicated towards the bibliography.

Historical background of CSV and XLS

CSV, or Comma-Separated Values, is a well known file type for data storage, much like XL, or Excel Binary File Format, files in the sense that both represent tabular data, however, the major difference is that CSV is a plain text file, each line representing a row and each comma a column, thus, anything between commas, a cell. CSV has a great historical background as it predates the personal computer being supported by the IBM Fortran under ‘OS/360‘ in 1972. The file type CSV itself, however, only came into existance in 1983.
XLS however only came into existance in 1987 with Excel's first Windows version, that being Excel 2. In 2007, however, XLS was deprecated and the use of XML versions of Excel spreadsheets was incentivized, thus introducing XLSX.

Historical background of JavaScript and JSON

JavaScript is a programming language famous for being one of the only capable of being used on the World Wide Web on the client side for webpage behaviour. It was first released in 1995 as an attempt to embed Java and Scheme

(a very popular LISP dialect), but was decided it was best to create a new language in itself with syntax more similar to Java than to Scheme.
Most importantly for this report is a specific data structure included in JavaScript, ‘objects‘, which are synonymous to assossiative arrays in other languages, thus, in the early 2000s, out of a need for stateless, real time server-to-browser communication protocols without Flash or Java applets, JSON files were created, these are code independant, as almost any language can parse it, but due to it’s inspired syntax from JavaScript, it’s is most popular for use with this programming language.

Importance of this project

Given the historical and practical significance of these three file formats, it is understandable how important it is to have a program that can seamlessly convert a CSV file into a JSON file, or, perhaps even, a program that converts a XLS file into a CSV file and thus allow for the previous program to convert it into JSON, that is what we aim to achieve with this project.
Again, considering how readable and easy to produce CSV files are, combined with out practical JSON files are, we can, from a plain text, send data in the text file from a server to a client, or display it on a web page.

Background of the Project

In this class, it was asked to solve one of five questions, the fifth, and the one we chose, is the conversion of a modified CSV format that includes lists and aggregation functions into a JSON file using regular expressions. These lists can be of fixed size N or a size between N and M, the aggregation functions were left at the students criteria.

Expansions of the Project

Considering XLS is nothing more than a zipped file containing XML files, this conversion should be trivial provided the ‘renaming‘ to the zipped file can be done, after which, theoretically, we should only need to find the cell data we need in the files inside the ‘xl‘ directory.
Indeed, this project only requires the conversion of a modified CSV into JSON, however, considering the properties CSV and XLS share, it seemed wise to at least explore the possibilities for the previously mentioned conversion, XLS variations into CSV.
In reality, this endeavor is not as easy as it might appear due to how obfuscated excel cell data is in the xml files that constitute it, however it is still an interesting topic that, despite it there being a functionality in Excel itself, a Python script that could convert XLS to CSV and perform the script created for the project would be of great importance.

1.2 Methodology

1.2.1 Theorical Backcground

Regular Expressions

A regular expression are a fucral component of Computer Science, both theoretical and practical, this because they were originated in the context of:

- Automata Theory
- Formal Languages

Because regular expressions express regular languages, we can use these for ‘pattern matching‘ allowing an user to easily find the first instance or all instances of a given pattern, or instead to replace one, all instances or a given amount of matches, this allows for ease of use for various actions such as, converting file types correctly, converting from a formal language into machine language, sorting and managing data and much more.

Python’s re module

In order to work with regular expressions, Python has a built-in module called ‘re‘ which allows the use of some powerful functions that take a raw string containing a regular expression, a string to be analyzed and produce, very efficiently, the desired result. The most important functions that will be used in this project are:

- re.search(regex,string)
- re.split(regex,string)
- re.sub(regex,regex,string)
- re.subn(regex,regex,string,count=n)

1.2.2 Practical component

Decisions

1.3 Conclusion

Chapter 2

Code