Processamento de Linguagens
Engenharia Informática (3º ano)
Projeto Final

23 de Março de 2023

Dispõe de **7 semanas** para desenvolver este trabalho, a entrega deverá ser feita até à meia noite de **14 de Maio**.

1 Objectivos e Organização

Este trabalho prático tem como principais objectivos:

- aumentar a experiência em engenharia de linguagens e em programação generativa (gramatical), reforçando a capacidade de escrever gramáticas, quer independentes de contexto (GIC), quer tradutoras (GT);
- desenvolver processadores de linguagens segundo o método da tradução dirigida pela sintaxe, a partir de uma gramática tradutora;
- desenvolver um compilador gerando código para um objetivo específico;
- utilizar geradores de compiladores baseados em gramáticas tradutoras, concretamente o Yacc, versão PLY do Python, completado pelo gerador de analisadores léxicos Lex, também versão PLY do Python.

Na resolução dos trabalhos práticos desta UC, aprecia-se a imaginação/criatividade dos grupos em todo o processo de desenvolvimento!

Deve entregar a sua solução até Domingo dia 14 de Maio. O ficheiro com o relatório e a solução deve ter o nome 'pl2023-projeto-grNN', em que NN corresponderá ao número de grupo. O número de grupo será ou foi atribuído no registo das equipes do projeto.

Cada grupo, **é livre** para escolher qual o enunciado que pretende desenvolver.

A submissão deverá ser feita por email com os seguintes dados:

to: jcr@di.uminho.pt

subject: PL2023::grNN::Projeto::Enunciado

body: Colocar um ZIP com os ficheiros do Projeto: relatório, código desenvolvido e datasets de teste.

Em cima, "Enunciado"é um dos valores: Pandoc, Ply-Simple, RecDesc, TDTabela.

Na defesa, a realizar na semana de 16 a 20 de Maio, o programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, em data e hora a marcar. O relatório a elaborar, deve ser claro e, além do respectivo enunciado, da descrição do problema, das decisões que lideraram o desenho da solução e sua implementação, deverá conter exemplos de utilização (textos fontes diversos e respectivo resultado produzido). Como é de tradição, o relatório será escrito em LaTeX.

2.6 Conversor toml-json

Ver detalhes em: https://github.com/toml-lang/toml

A linguagem *toml* permite uma fácil definição de estruturas complexas (dicionários generalizados) frequentmente usados em ficheiros de configuração e em vários outros domínios de modo análogo ao JSON e ao YAML

Considere o sequinte exemplo:

```
title = "TOML Example"

[owner]
name = "Tom Preston-Werner"
date = 2010-04-23
time = 21:30:00

[database]
server = "192.168.1.1"
ports = [ 8001, 8001, 8002 ]
connection_max = 5000
enabled = true

[servers]
[servers.alpha]
  ip = "10.0.0.1"
  dc = "eqdc10"

[servers.beta]
  ip = "10.0.0.2"
  dc = "eqdc10"

# Line breaks are OK when inside arrays
hosts = [
  "alpha",
  "omega"
]
```

Pretende-se construir uma ferramenta (Flex,Yacc) que converta um subconjunto de Toml para JSON.