

UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO (UFERSA)
CENTRO DE CIÊNCIAS EXATAS E NATURAIS (CCEN)
DEPARTAMENTO DE COMPUTAÇÃO
DISCIPLINA: COMPILADORES
ATIVIDADE: TRABALHO PRÁTICO SOBRE ANÁLISE LÉXICA

OBJETIVO: Construir um analisador léxico para reconhecimento de tokens da linguagem OWL2 (Web Ontology Language) no formato Manchester Syntax.

DESCRIÇÃO: A linguagem **OWL** (*Web Ontology Language*) é baseada em **RDF** (*Resource Description Framework*) e **XML** (**eXtensible Markup Language**) para relacionar recursos que têm algo a ver um com o outro na Web. Tudo na Web é um recurso: uma página, um perfil de utilizador de redes sociais, um vídeo, um áudio, um texto, entre outros. O nível de granularidade dos recursos na Web é vasto. Cada recurso pode ser identificado por uma **URI** (*Unified Resource Identifier*). Uma **URL** (*Unified Resource Locator*) é um tipo específico de URI que identifica um endereço único para um determinado recurso na Web. Uma ontologia é um vocabulário que descreve conceitos de uma determinada área do conhecimento. Se esses conceitos forem materializados como recursos na Web, então uma ontologia pode estabelecer relações semânticas entre esses conceitos. Ontologias e dados abertos conectados (*Linked Data*) são as estruturas de dados que ajudam motores de busca, sites de comércio eletrónico e redes sociais a conectarem recursos que têm a ver um com o outro na Web. Agentes inteligentes e serviços Web desenvolvidos com APIs baseadas em Lógica de Descrição (*Description Logics*) podem inferir conhecimento novo a partir de ontologias e dados abertos conectados, e recomendá-lo aos utilizadores de diversos serviços na Web.

OWL é uma linguagem para inferência de máquina, embora ontologistas possam compreendê-la. Abaixo, podemos ver um trecho de uma ontologia de pizzas, o qual descreve as características de uma pizza *Margherita*:

```
<!-- http://www.co-ode.org/ontologies/pizza/pizza.owl#Margherita -->
<owl:Class rdf:about="http://www.co-ode.org/ontologies/pizza/pizza.owl#Margherita">
  <rdfs:subClassOf rdf:resource="http://www.co-ode.org/ontologies/pizza/pizza.owl#NamedPizza"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.co-ode.org/ontologies/pizza/pizza.owl#hasTopping"/>
      <owl:someValuesFrom
        rdf:resource="http://www.co-ode.org/ontologies/pizza/pizza.owl#MozzarellaTopping"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.co-ode.org/ontologies/pizza/pizza.owl#hasTopping"/>
      <owl:someValuesFrom
        rdf:resource="http://www.co-ode.org/ontologies/pizza/pizza.owl#TomatoTopping"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.co-ode.org/ontologies/pizza/pizza.owl#hasTopping"/>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <rdf:Description
              rdf:about="http://www.co-ode.org/ontologies/pizza/pizza.owl#MozzarellaTopping"/>
            <rdf:Description
              rdf:about="http://www.co-ode.org/ontologies/pizza/pizza.owl#TomatoTopping"/>
          </owl:unionOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```

        </owl:unionOf>
    </owl:Class>
</owl:allValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:label xml:lang="en">Margherita</rdfs:label>
<rdfs:label xml:lang="pt">Margherita</rdfs:label>
<skos:altLabel xml:lang="en">Margherita</skos:altLabel>
<skos:altLabel xml:lang="en">Margherita Pizza</skos:altLabel>
<skos:prefLabel xml:lang="en">Margherita</skos:prefLabel>

```

Esse mesmo trecho de ontologia pode ser descrito de forma mais palatável para o utilizador humano no formato Manchester Syntax, para descrição de ontologias em Description Logics:

```

Pizza THAT
hasTopping SOME MozzarellaTopping AND
hasTopping SOME TomatoTopping AND
hasTopping ONLY (MozzarellaTopping OR
TomatoTopping OR
PepperonniTopping)

```

Description Logics é a lógica de descrever coisas com base nas relações que elas têm entre si. Uma linguagem baseada em Description Logics é geralmente declarativa, ou seja, não contém métodos ou funções de transformação. Sendo assim, os conceitos de uma ontologia, por exemplo, são descritos sequencialmente, como declarações avulsas. Um motor de inferência (*reasoner*) lê essas declarações e realiza inferências sobre relações implícitas que poderiam conectar ainda mais os conceitos de uma ontologia. Poderíamos também descrever outros conceitos da ontologia usando o mesmo tipo de lógica, por exemplo:

```
<!-- descreve algo que é pizza e tem pelo menos 3 tipos de cobertura -->
```

```

Pizza THAT
hasTopping min 3

```

```
<!-- descreve algo que é pizza e tem mais de 400 calorias -->
```

```

Pizza THAT
hasCaloricContent some integer[>="400"]

```

```
<!-- descreve algo que é pizza mas não é uma pizza vegetariana -->
```

```

Pizza THAT
not VegetarianPizza

```

```
<!-- descreve algo que é uma cobertura de pizza, que é cobertura de queijo, que é
levemente apimentado e tem a Itália como país de origem -->
```

```

PizzaTopping AND
CheeseTopping THAT
hasSpiciness SOME Mild
and hasCountryOfOrigin VALUE Italy

```

DESAFIO: Especificar um analisador léxico para a linguagem OWL2 no formato Manchester Syntax, considerando as seguintes especificações:

Palavras reservadas:

- *SOME, ALL, VALUE, MIN, MAX, EXACTLY, THAT*
- *NOT*
- *AND*
- *OR*

Identificadores de classes:

- Nomes começando com letra maiúscula, p.ex.: *Pizza*
- Nomes compostos concatenados e com iniciais maiúsculas, p.ex.: *VegetarianPizza*
- Nomes compostos separados por *underline*, p.ex.: *Margherita_Pizza*

Identificadores de propriedades:

- Começando com “has”, seguidos de uma string simples ou composta, p.ex.: *hasSpiciness*, *hasTopping*, *hasBase*.
- Começando com “is”, seguidos de qualquer coisa, e terminados com “Of”, p.ex.: *isBaseOf*, *isToppingOf*.

Cardinalidades:

- Representadas por números inteiros, p.ex.: *hasTopping min 3*

O analisador léxico deve produzir como saída os tokens de cada elemento da linguagem especificados em uma tabela de símbolos.

FERRAMENTAS:

- Especificação manual, usando NFAs, DFAs, transformações NFA→DFA e expressões regulares
- Ambientes de desenvolvimento integrado, p.ex.: VS Code
- Ambientes específicos de modelagem para compiladores, p.ex.: Flex ou ANTLR

OBSERVAÇÕES IMPORTANTES:

- O trabalho corresponderá a **70%** da nota da **Unidade 1**
- Poderá ser realizado **individualmente** ou em grupos de **2 componentes**
- Data de entrega: **21/12/2023**, até às **23h59**

REFERÊNCIAS

Horridge, M., Drummond, N., Goodwin, J., Rector, A. L., Stevens, R., & Wang, H. (2006, November). The Manchester OWL syntax. In *OWLed* (Vol. 216). Disponível online em: https://ceur-ws.org/Vol-216/submission_9.pdf

Parr, T. (2013). The definitive ANTLR 4 reference. *The Definitive ANTLR 4 Reference*, 1-326.

McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, 10(10), 2004. Disponível online em: <https://static.twoday.net/71desa1bif/files/W3C-OWL-Overview.pdf>