

MAC0444

Relatório do Projeto

André Ferrari Moukarzel - 9298166
Fábio Henrique Kiyoyiti dos Santos Tanaka - 9297982
Gabriel Kuribara Lasso - 9298016
Vitor Kei Taira Tamada - 8516250

Conceitos:

1. **Actor** (ator(a))
2. **Director** (diretor(a))
3. **Movie** (filme)

Propriedades:

Actor:

- Subclasse de **Person**
- *actorOf*:
Relaciona-se com uma instância de **Movie**. Se um ator atuou em um filme, ele terá a relação *actorOf* com este filme. Um ator pode possuir uma relação deste tipo com múltiplos filmes.
Relação inversa a *actedBy*.

- *firstName* e *familyName*:
Primeiro nome e sobrenome do indivíduo respectivamente.

Director:

- Subclasse de **Person**
- *directorOf*:
Relaciona-se com uma instância de **Movie**. Se um diretor dirigiu um filme, ele terá a relação *directorOf* com este filme. Um diretor pode possuir uma relação deste tipo com múltiplos filmes.
Relação inversa a *directedBy*.
- *firstName* e *familyName*:
Primeiro nome e sobrenome do indivíduo respectivamente.

Movie:

- Subclasse de **Project**
- *actedBy*:
Relaciona-se com uma instância de **Actor**. Se um ator atuou no filme em questão, o filme terá a relação *actedBy* com tal ator. Um filme pode possuir uma relação deste tipo com múltiplos atores.
Relação inversa a *actorOf*.
- *directedBy*:
Relaciona-se com uma instância de **Director**. Se um diretor dirigiu o filme em questão, o filme terá a relação *directedBy* com tal diretor. Um filme pode possuir uma relação deste tipo com múltiplos diretores.
Relação inversa a *directorOf*.
- *releaseYear*:
Número inteiro que representa o ano de lançamento do filme. Filmes com data de lançamento desconhecida possuem o ano 0 neste campo.

Instâncias:

A princípio, adicionamos Uma Thurman, Harvey Keitel, Bill Murray e Frances McDormand como atores (**Actor**), e Quentin Tarantino e Wes Anderson como diretores (**Director**). Então, pegamos todos os filmes e séries dos quais estes participaram. A partir desse ponto, adicionamos todos os outros atores e diretores envolvidos nessas mesmas séries e filmes.

Para a criação do arquivo em OWL, criamos instâncias de filmes com as relações apropriadas a todos os atores e diretores neles envolvidos, e uma instância para cada ator e diretor sem relação alguma. Disso, inferimos no Protégé as relações restantes dos atores e diretores com os filmes.

Consultas:

Tendo o prefixo “my” como sendo nossa ontologia.

1) Quais atores participaram do filme F?

```
SELECT ?actor
WHERE {?actor my:actorOf my:F}
```

2) Quais filmes foram dirigidos pelo diretor D?

```
SELECT ?movie
WHERE {?movie my:directedBy my:D}
```

3) Em quais filmes o ator X atuou?

```
SELECT ?movie
WHERE {?movie my:actedBy my:X}
```

4) Em quais filmes o ator X atuou junto com Y?

```
SELECT ?movie
  WHERE {
    ?movie my:actedBy my:X .
    ?movie my:actedBy my:Y
  }
```

5) Quem foram os diretores dos filmes nos quais os atores X e Y atuam juntos?

```
SELECT ?director
  WHERE {
    ?director my:directorOf ?movie .
    ?movie my:actedBy my:X .
    ?movie my:actedBy my:Y
  }
```

6) Qual o diretor que mais dirigiu filmes do ator X?

```
SELECT ?director (COUNT(?movie) as ?quantidade)
  WHERE {
    ?movie my:actedBy my:X .
    ?director my:directorOf ?movie
  }
GROUP BY ?director
ORDER BY DESC (?quantidade)
LIMIT 1
```

7) Qual o ator que mais aparece nos filmes do diretor D?

```
SELECT ?actor (COUNT(?movie) as ?quantidade)
  WHERE {
    ?movie my:directedBy my:D .
    ?actor my:actorOf ?movie
  }
  GROUP BY ?actor
  ORDER BY DESC (?quantidade)
  LIMIT 1
```

8) Entre os anos N1 e N2, quais diretores dirigiram filmes onde X e Y aparecem?

```
SELECT ?director
  WHERE {
    ?movie my:actedBy my:X .
    ?movie my:actedBy my:Y .
    ?movie my:directedBy ?director .
    ?movie my:releaseYear ?date
    FILTER (
      ?date > N1 &&
      ?date < N2
    )
  }
```

9) Entre os anos N1 e N2, quais atores atuaram juntos nos filmes onde X e Y aparecem?

```
SELECT ?actor
  WHERE {
    ?movie my:actedBy my:X .
    ?movie my:actedBy my:Y .
    ?movie my:actedBy ?actor .
    ?movie my:releaseYear ?date
    FILTER (
      ?date > N1 &&
      ?date < N2 &&
      ?actor != my:X &&
      ?actor != my:Y
    )
  }
```

10) Quais filmes do diretor do filme F possuem X ou Y como atores?

```
SELECT DISTINCT ?movie
  WHERE {
    ?director my:directorOf my:F .
    ?director my:directorOf ?movie .
    ?actor my:actorOf ?movie .
    FILTER ( ?actor = my:X || ?actor = my:Y)
  }
```

Outra forma de fazer a consulta 10 seria:

```
SELECT DISTINCT ?movie
  WHERE {
    ?director my:directed my:F .
    ?director my:directed ?movie .
    {
      ?movie my:actedBy my:X
    } UNION {
      ?movie my:actedBy my:Y
    }
  }
```

Resultados:

A ontologia obtida após as inferências é consideravelmente grande. Embora não tenha sido necessário remover as séries da base de dados para sermos capazes de abri-la no Protégé, qualquer forma de manipulação é extremamente lenta, incluindo consultas em SPARQL.

Portanto, para o bem dos computadores da Rede Linux do IME, testamos as consultas em uma base de dados artificial que criamos manualmente, com apenas algumas instâncias de atores, diretores e filmes. Nela as operações foram realizadas de forma bem rápida e devolveram os resultados esperados, porém essa eficiência se deu por conta do pequeno tamanho da base de conhecimento e não pela eficiência do programa ou da ontologia.

Vale mencionar que, com uma base de dados que possuía todos os filmes e séries dos atores e diretores requisitados, incluindo suas relações, filmes e as pessoas com quem trabalharam, a indução no Protégé pelas relações restantes levou cerca de 2 minutos.