

**Relatório KRR (Representação de conhecimento) - (2018-1)**

**Elder Rizzon Santos**

Professor do Depto. de Informática e Estatística da Universidade Federal de Santa Catarina

**Fabio Nunes Garcia (15100733)**

**Gustavo Figueira Olegário (15100742)**

Alunos da disciplina INE5430 - Inteligência Artificial

Ciência da Computação do Depto. de Informática e Estatística da Universidade Federal de  
Santa Catarina

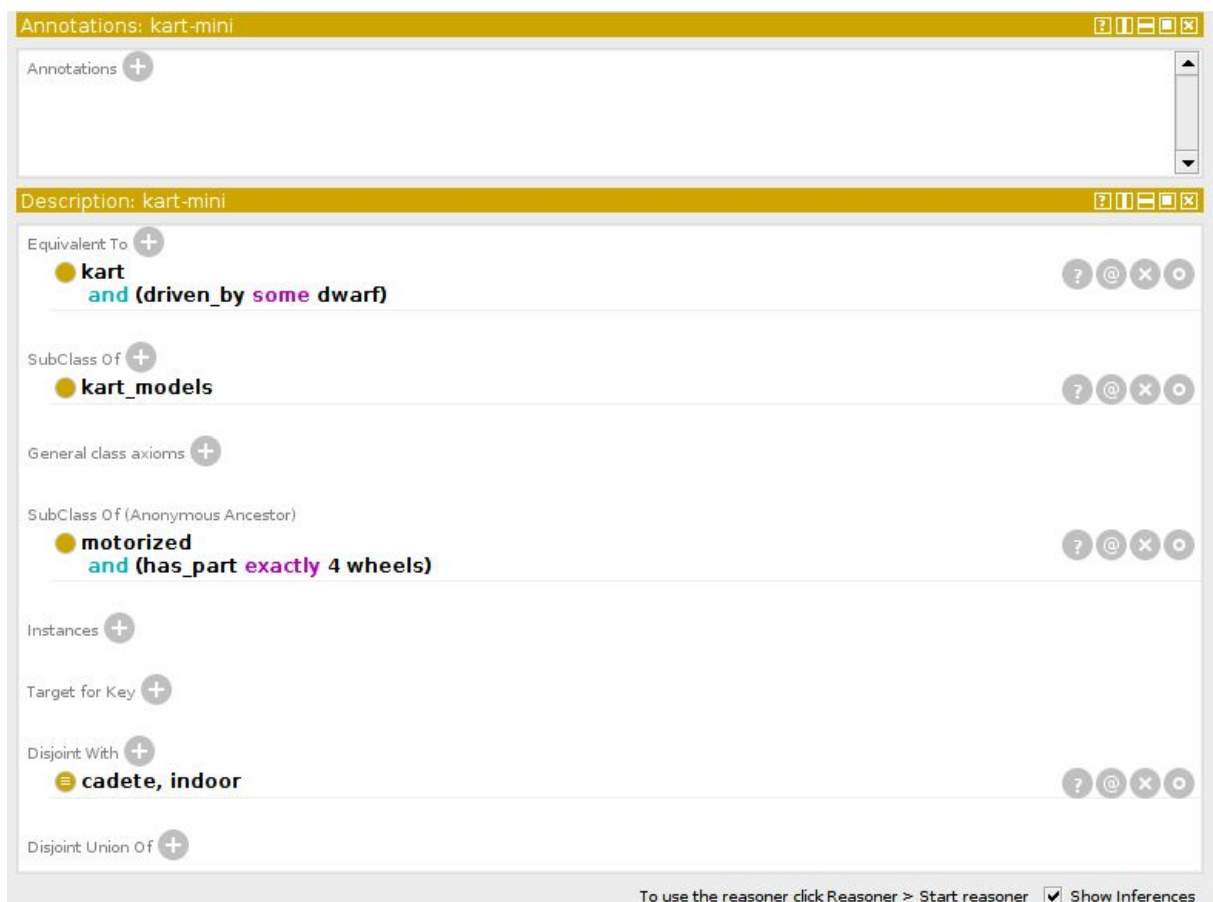
Florianópolis, 13 de Abril de 2018.

## 1. Introdução

Este relatório busca responder todas as perguntas solicitadas no segundo trabalho prático da disciplina INE 5430 no primeiro semestre de 2018, bem como representar a modelagem desenvolvida pelo grupo na linguagem OWL.

## 2. Parte 1: pesquisa teórica

- a) A diferença semântica entre `subClassOf` e `equivalentTo` se dá pelo fato de que toda classe que for declarada e tiver a palavra `subClassOf` deverá, obrigatoriamente, herdar todas as características da classe mãe. Ou seja Se uma classe B for subclasse de outra A e essa, por sua vez, tiver uma propriedade X qualquer, então necessariamente B também deverá ter essa mesma propriedade X e, opcionalmente, pode ter suas próprias características. Porém, se implementarmos uma nova classe C e definirmos ela como sendo equivalente a classe B, com suas próprias particularidades, qualquer instância de C que respeitar as características pré-definidas, também será avaliada como uma instância de B, ainda que as particularidades de C nada tenham a ver com o que foi definido em B. Abaixo temos uma captura de tela do projeto proposto pela dupla:



Nessa classe denominada "kart-mini" definiu-se que ela é uma subclasse "kart\_models". Ou seja, ela herdou todas as características da classe mãe. Por outro lado, foi definido que ela é também equivalente a classe "kart"

b) Sabemos que a lógica descritiva, OWL, é um problema computacional do tipo decidível. Em outras palavras, sempre que qualquer programa escrito em OWL for executado ele terá parada garantida. Isso se deve ao fato de que o interpretador sempre terá que devolver uma resposta ao usuário, ainda que seja algo como: "Pode ser que exista". Como estamos no universo de problemas computáveis, sabemos que todo problema computável tem uma máquina de Turing associada. Esse autômato, por sua vez, é perfeitamente descrito com lógica de primeira ordem. Porém, os problemas computáveis conseguem representar os problemas decidíveis e não decidíveis, já que é possível representar o problema da parada apenas com máquinas de Turing, problema esse que é naturalmente indecidível. Dessa forma fica provado que a lógica de primeira ordem é muito mais expressiva que a OWL, uma vez que essa não é capaz de representar problemas indecidíveis

Já quanto ao aspecto prático deste trabalho, foi implementado um conjunto de classes e propriedades envolvendo veículos e seus respectivos perfis de usuários. Inicialmente, foram desenvolvidas classes para representar veículos e suas restrições de usuário como por exemplo: um carro do modelo Civic só pode ser dirigido por senhores de idade e precisa, necessariamente, ter 4 rodas, enquanto que uma Duster é pilotada por um homem que possua barba e também atenda à condição do número de rodas mencionadas anteriormente.

Além disso, as montadoras de cada carro estão representadas nessa ontologia, juntamente com a localização das mesmas em cada país. Ainda sobre as classes e suas representações, existe a relação de que cada país produz ao menos um tipo de comida: arroz, milho, feijão, trigo. Quanto às relações de equivalência definiu-se como carros sendo equivalentes a karts. No que se refere às propriedades, cada veículo pode ter uma propriedade que o defina unicamente. Por exemplo: a propriedade `has_great_motor`, é uma propriedade em que o domínio são carros somente do tipo camaro.

#### **Referências:**

- <https://stackoverflow.com/questions/4192435/owls-equivalentclass-vs-subclassof>
- [https://pt.wikipedia.org/wiki/Problema\\_de\\_decis%C3%A3o](https://pt.wikipedia.org/wiki/Problema_de_decis%C3%A3o)
- [https://pt.wikipedia.org/wiki/Teoria\\_da\\_computabilidade](https://pt.wikipedia.org/wiki/Teoria_da_computabilidade)