ADEETC - Mestrado em Engenharia Informática

Guia Aula Prática

Representação e Processamento de Conhecimento (RPC)

1. Instalação e versão (a utilizar) do Protégé (para "Desktop")

Vamos usar o Protégé (`http://protege.stanford.edu/`). Que versão "Protégé Desktop" usar? As mais recentes são: 3.5 (24.abr.2013), 4.3 (15.abr.2013) e a versão 5.2.0 (15.mar.2017); cf., `http://protegewiki.stanford.edu/wiki/Protege_Desktop_Old_Versions/` e `https://protegewiki.stanford.edu/wiki/Protege-OWL`.

Porquê usar o Protégé versão 4.x e não a versão 3.x? Há uma comparação bastante completa em: `http://protegewiki.stanford.edu/index.php/Protege4Migration`.

Essa comparação tem aspetos desatualizados. Indica que a versão 4.0 ainda não suporta SPARQL mas não diz que a 4.1 já suporta SPARQL. As versões 4.x têm integração direta com os motores de inferência (e.g., FaCT++); as versões 3.x usam motor de inferência como serviço externo. As 4.x já implementam o OWL 2.0. Assim, vamos usar o Protégé versão ≥ 4.1. Em 2013/14 usámos a versão 4.3; em 2014/15 Protégé 5.0 versão beta; 2015/16 o Protégé 5.0 (*release*); 2016/17 Protege 5.2.0.

Este ano 2017/18 continuamos com **Protégé 5.2.0**; em "`./distribuicao_05`" (versão Windows).

Para usar o Protégé 5.2.0 basta descomprimir o ficheiro, abrir uma janela de consola, mover para a pasta gerada ("`Protege-5.2.0-win\Protege-5.2.0`") e aí executar "`run.bat`" (convém executar numa janela de consola para conseguir ver as mensagens de erro e outras).

No entanto, no Windows10 tive um erro indicando que "a versão de Java usada pelo Protégé não executava na versão do sistema operativo!". Para ultrapassar fiz o seguinte: (a) renomeei a pasta "`Protege-5.2.0\jre`" para "`Protege-5.2.0\jre_ori`", (b) acedi à pasta do jre instalado na máquina (no meu caso em "`C:\Program Files\Java\jre1.8.0_161`" e copiei a pasta "`jre1.8.0_161`" para "`Protege-5.2.0`" e renomeei "`jre1.8.0_161`" para "`jre`". Ou seja, substitui o jre "bundled" com o Protégé pelo jre instalado na minha máquina. Olhe para o código de "`run.bat`" para perceber como é iniciada a execução do Protégé.

Assim, e contexto deste guião tudo foi testado num sistema Windows 10, 32bits; e com java (JRE) "1.8.0_161". Para ver qual a versão de java fazer (em "*Command Prompt*": `java -version`).

Em geral, se tiver várias versões de Java poderá indicar qual pretende usar. No Windows (<10) normalmente será "`C:\WINDOWS\system32\java.exe`"; o primeiro no "PATH"; para ver qual o seu "path" fazer (em "*Command Prompt*": `echo %PATH%`). No entanto esta versão do Protégé usa explicitamente a versão do Java contida no seu pacote pelo que foi preciso fazer o ajuste acima descrito para que passasse a funcionar no Windows10.

Toda a documentação está disponível em:
`http://protege.stanford.edu/support.php`.
`http://protegewiki.stanford.edu/wiki/Main_Page`.
`http://protegewiki.stanford.edu/wiki/Protege-OWL_4_FAQ`.

Em alguns aspectos a documentação apresenta grafismo (e.g., imagens de janelas) que podem não ser exatamente iguais às que correspondem à versão que está a usar; e.g., alguns documentos

assumem a versão 3.3.1 (essa foi uma versão de referência na construção da documentação do Protégé). No entanto, em geral é fácil entender o que se pretende e fazer o corresponde ajuste.

Atenção: ao utilizar uma versão anterior do Protégé detectei o seguinte cenário de erro: "ao adicionar um individuo (separador "*Individuals by class*") estando o "reasoner" FaCT++ activo a aplicação terminava de modo abrupto"; este problema não ocorreu com o "reasoner" HermiT 1.3.7; por omissão este é o único "reasoner" na versão 5.0.0.

Conclusão: quando precisar de usar o "reasoner" esteja atento a este cenário para o FaCT++ ou utilize o HermiT 1.3.7 (ou o Pellet que será adicionado no decurso desta aula).

2. Verificar Protégé e instalar o GraphViz

a) Executar o Protégé (executar "`run.bat`" na pasta construída); em resposta à janela para atualização de "plugin" escolhi "Not now".

b) Escolha (no menu) a opção "Window\Tabs\OWLViz" e selecione OWLViz. **Se tiver mensagem de erro** (na janela de consola) relacionada com a inexistência de uma aplicação "DOT" fazer próximos passos. Nesta versão 5.2.0 não obtive este erro. **Caso também não tenha mensagem de erro** continuar para o próximo exercício.

c) Instalar o "GraphViz" (`http://www.graphviz.org/Download_windows.php`); a versão 2.38, que funcionou desde o Protégé 5.0.0; em `./distribuicao_05`. ***Atenção***: para **Windows Vista** leia o "***Warning***" em `http://www.graphviz.org/Download_windows.php` e ***analise as indicações*** aí propostas; este ano a informação está inativa; Windows Vista descontinuado.

d) Para instalar basta executar. No processo de instalação tenha atenção ao local ("path") onde indica que pretende colocar o "GraphViz" pois precisará de configurar o Protégé para esse local.

e) Depois de instalar o "GraphViz" execute o Protégé e vá a "File\Preferences..."; aí escolha o separador "OWL Viz"; em "Path" coloque todo o caminho para a aplicação "`dot.exe`"; por exemplo "`C:\myApp\Graphviz2.38\bin\dot.exe`" (se tiver sido instalado em "`C:\myApp`").

f) Volte a testar escolhendo o separador "OWL Viz". Caso o problema se mantenha veja mais detalhe em: `http://protegewiki.stanford.edu/wiki/Protege-OWL_4_FAQ`.

3. Adicionar separadores ("tab") e "fazer-experiência com SPARQL"

a) Executar o Protégé ("`run.bat`") e escolha (no menu) "Window\Tabs\" e depois selecione:
\"Object Properties"
\"Data Properties"
\"OntoGraf"
\"SPARQL Query".

b) Selecionar o separador "Entities" e adicionar a classe "A" (escolher ícone no topo-esquerdo da janela "Class hierarchy"); selecionar a classe "A" e criar classe "B" (fica sub-classe de "A").

c) Escolher o separador "SPARQL Query" e executar a interrogação de omissão. Caso não surja uma interrogação de omissão escolha: "<Window \ Reset selected tab to default state>". Caso a

janela continue vazia instale os "plugins": "OWLAPI RDF Library" e "Protege SPARQL Plugin" (cf., `https://github.com/protegeproject/protege/issues/636`); para instalar os plugins escolha "<File \ Check for plugins...>". Reinicie o Protégé para ativar os plugins. Caso a janela continue vazia volte a escolha: "<Window \ Reset selected tab to default state>". Por fim, quando funcionar analise o resultado da interrogação de omissão.

d) Escolher "\File\Save as...", escolher o formato "RDF/XML Syntax" e gravar o ficheiro com nome "myFirstTBox"; analise o ficheiro gerado.

e) Execute (no Protégé) uma diretiva SPARQL para obter os sujeitos do tipo "owl:Class" (algo do género "`?subject rdf:type owl:Class`"). Note no resultado 2 "A"s e 2 "B"s; porquê?

f) Selecione o separador "Entities", selecione a classe "B" e arraste para cima de "Thing" (agora "A" e "B" são classes irmãs). Executar a diretiva SPARQL da alínea anterior. Agora tem 1 "A" e 1 "B".

g) Pode gravar esta nova ontologia e depois selecione "File\New" e <Yes> (na mesma janela).

Daqui para a frente vamos seguir um tutorial mas vamos "intercalar" com um exemplo nosso.

ADEETC - Mestrado em Engenharia Informática

Guia Aula Prática

Representação e Processamento de Conhecimento (RPC)

4. Criar uma nova Ontologia e atribuir Comentário (usando guião)

Ao longo desta aula será usado o documento "`01_ProtegeOWLTutorialP4_v1_3.pdf`". No entanto, na pasta da aula estão também dois documentos adicionais como proposta para leitura complementar "`02_ontologyDevelopment.pdf`" e "`03_OWLVizGuide.pdf`".

Abra o tutorial "`01_ProtegeOWLTutorialP4_v1_3.pdf`" no capítulo 4. Os capítulos anteriores abordam os conceitos teóricos portanto é também importante lê-los (depois da aula prática!). Este tutorial baseia-se na interface gráfica de uma versão anterior do Protégé. Portanto leia a secção 4.1 ("*Exercise 2 e 3*; pág. 13-15 do tutorial") mas siga as instruções nas próximas alíneas (aqui abaixo!).

a) Execute o Protégé.

b) No separador "Active Ontology", em "Ontology IRI" indique: `http://www.pizza.com/ontologies/pizza.owl`.

c) Em "File\Save as..." escolha "RDF/XML" e grave o ficheiro "`pizza.owl`" na sua pasta de trabalho.

d) No separador "Active Ontology", escolha "Annotations" e carregue em "(+)", depois "comment" e atribua o comentário sugerido no "*Exercise 3*, do tutorial"

Note que este exercício corresponde aos "*Exercise 2 e 3*; pág. 13-15 do tutorial"

5. Construir Conceitos e Axiomas de Inclusão (usando guião)

Abra o tutorial "`01_ProtegeOWLTutorialP4_v1_3.pdf`" no capítulo 4. Os capítulos anteriores abordam os conceitos teóricos portanto é também importante lê-los (depois da aula prática!).

a) *Atenção*: deve selecionar o separador "Entities" (por omissão fica no "Active Ontology"). Construa os conceitos (classes) seguindo as indicações até à secção 4.2. Note que a diferença entre as figuras no documento e as janelas desta versão do Protégé são apenas no grafismo.

b) Altere o nome de um dos conceitos que construiu. Para isso selecione o conceito que pretende renomear e vá a: "Refactor \ Rename entity...". Volte a repor o nome original desse conceito.

c) Leia com atenção a secção 4.2 (conceitos disjuntos) e construa a disjunção aí indicada.

d) Grave as alterações e leia o ficheiro "`pizza.owl`" (definido no exercício anterior).

e) Na secção 4.3 veja como construir axiomas de inclusão (relações classe-subclasse); ***note*** que o Protégé permite construir rapidamente várias classes ("Tools\Create class hierarchy...").

f) Leia o "*Exercise 7*", e construa toda a hierarquia da figura 4.10 de uma única vez.

g) Não prosseguir, por enquanto, para a secção 4.4; i.e., ficar na página 23 e gravar o projecto.

ADEETC - Mestrado em Engenharia Informática

Guia Aula Prática

Representação e Processamento de Conhecimento (RPC)

6. Construir Axiomas de Igualdade (sem usar guião)

Grave o projecto anterior.

a) ... e faça "\File\New..."; na janela de diálogo "*Do you want to open the ontology in the current window*?" responda <No>.

b) Dê o nome "`http://FAMILA`" à nova ontologia que está a construir e grave em "`familia.owl`".

c) Considere a TBox em "`a04_logicaDeDescricao.pdf`", página 25 (relações de parentesco).

d) Construa apenas os conceitos (classes): `Pessoa`, `Feminino`, `Mulher`, `Homem`. Construa todas de uma vez ("Tools\Create class hierarchy...."); note a indicação sobre se são ou não disjuntas!

e) Construa os dois primeiros axiomas de igualdade da TBox (indicados nas alíneas i e ii). Recorde um axioma de igualdade é uma condição de equivalência; bloco "`Equivalent To`" no Protégé. Assim, deve seleccionar o conceito que corresponde ao lado esquerdo da definição e no bloco "`Equivalent To`" picar no símbolo ("+"). A janela que surge tem o nome do conceito que está do lado esquerdo do axioma; nessa janela, no separador "Class expression editor" escrever o lado direito do axioma; para i) `Pessoa and Feminino`, e para ii) `Pessoa and not Mulher`. Com <ctrl-space> e, por vezes <tab>, faz "auto complete".

f) Para informação adicional sobre este "Class expression editor" ver:
`http://protegewiki.stanford.edu/wiki/Protege4ExpressionEditor`, e
`http://protegewiki.stanford.edu/wiki/Protege4UserDocs#Editor_features/`.

g) Grave e leia o ficheiro "`familia.owl`". Recorde a seriação de um grafo RDF(S) em XML.

h) Não construir, por enquanto, os restantes axiomas.

7. Construir Relações, ou Propriedades, ou Papéis (usando guião)

Grave o anterior e regresse à ontologia da pizza (deve estar noutra janela, já aberta, do Protégé).

Abra o "`01_ProtegeOWLTutorialP4_v1_3.pdf`" na secção 4.4 (pág. 23).

a) Siga o tutorial para construir as propriedades `hasTopping` e `hasBase` como sub-propriedades de `hasIngredient`. Note que nesta versão do Protégé os diversos tipos de propriedade ("object", "data", "annotation") correspondem a 3 separadores na janela principal (caso algum não esteja ativo já sabe como fazer em "Window\Tabs").

b) Leia com atenção a secção 4.5 (propriedades inversas) e construa as propriedades aí indicadas.

c) *Sugestão adicional*: instale "plug-in" "Matrix view" (em File\Check for plugins...) e depois ative (em Window\Tabs) a "Property matrix" pois simplifica manipulação das propriedades.

d) Não prosseguir, por enquanto, para a secção 4.6; i.e., ficar na página 28 e gravar o projeto.

ADEETC - Mestrado em Engenharia Informática

Guia Aula Prática

Representação e Processamento de Conhecimento (RPC)

8. Construir e "Configurar" Relações (sem usar guião)

Grave o anterior e regresse à ontologia "`http://FAMILA`".

a) Construa a relação `temFilho` (em "Object Properties") e indique que tem como domínio e contradomínio `Pessoa`; nos blocos "Domain" e "Range"

b) Veja com atenção a secção 4.6 do "`01_ProtegeOWLTutorialP4_v1_3.pdf`" e decida se a relação "`temFilho`" deve ser "`Functional`", "`Inverse Functional`", "`Transitive`", "`Symmetric`" "`Asymmetric`", "`Reflexive`", "`Irreflexive`" (estas noções são "check boxes" em "Characteristics" de "Object Properties").

c) Grave e leia o ficheiro "`familia.owl`".

d) Não construir, por enquanto, os restantes axiomas.

9. Construir e "Configurar" Relações (usando guião)

Grave o anterior e regresse à ontologia da pizza.

a) Para sistematizar, faça os exercícios das secções 4.6 e 4.7.

b) Não prosseguir, por enquanto, para a secção 4.8; i.e., ficar na página 36 e gravar o projeto

10. Usar o Motor de Inferência Pré-Configurado (FaCT++ e Hermit)

Grave o anterior e regresse, copie a ontologia da pizza e use, neste exercício, essa cópia.

O motores de inferência Hermit está perfeitamente integrados na atual versão do Protégé.

a) Seguir o tutorial na secção 4.9 considerando que nesta versão:

- não existe o botão "superclasses" portanto no "*Exercise 24*" em vez de indicar que "VegetableTopping" é superclasse de "ProbeInconsistentTopping" indica que "ProbeInconsistentTopping" é subclasse de "VegetableTopping"; i.e., carregar no sinal (+) em "*SubClass Of*" estando "ProbeInconsistentTopping" selecionado.
- a opção de menu "Reasoner" tem as alternativas: "Start reasoner", "Synchronize reasoner", "Stop reasoner" e "Explain inconsistente ontology".

b) Siga a secção 4.9 de modo a gerar a inconsistência aí descrita.

c) Qual a hierarquia de subsunção que se infere da TBox construída? Para obter a resposta selecionar "Class hierarchy (inferred)" e observar a taxonomia aí apresentada.

d) Elimine a inconsistência anterior editando "*Disjoint Union Of*".

11. Adicionar um Novo Motor de Inferência (FaCT++ e Pellet)

Grave o anterior e regresse à ontologia da pizza.

O motor de inferência Pellet é uma referência importante nesta área (foi pioneiro, tem API e também versão para Lógica Descrição Probabilística) pode integrar o Protégé versão 4.3 mas nesta versão 5.0 não tem ainda suporte. Vamos por isso adicionar o motor FaCT++ também pioneiro nesta área.

a) Selecionar "File \ Check for plugins..."; espere um pouco!

b) Escolher "Pellet Reasoner"; escolher "Install".

c) Reiniciar o Protégé (i.e., fazer "File\Exit e voltar a lançar o Protégé).

d) … agora tem dois motores de inferência; refazer exercício anterior agora com "reasoner" FaCT++.

Caso tenha erro de incompatibilidade entre o "reasoner" e o seu sistema o melhor é ir à pasta "`Protégé-5.2.0\plugins`" e remover o ficheiro `.jar` que corresponde ao plugin "problemático".

12. Especificar Restrições com Quantificadores (sem usar guião)

Grave o anterior e regresse à ontologia "`http://FAMILA`".

a) Construir o axioma iii (define `Mae`) da TBox. Para isso, construir o conceito de `Mae` (subclasse de `Thing`) e no bloco "`Equivalent To`" selecionar "+" e no "Class expression editor" escrever: "`Mulher and (temFilho some Pessoa)`". Notar os quantificadores escritos em notação infixa (na lógica de descrição usámos escrita prefixa). Mais informação sobre esta sintaxe em: `http://www.co-ode.org/resources/reference/manchester_syntax/`.

b) Construir o axioma iv (define `Pai`) da TBox.

c) Construir o axioma v (define `Progenitor`) da TBox.

d) Construir o axioma vi (define `Avó`) da TBox.

e) Grave e leia o ficheiro "`familia.owl`". Recorde a seriação de um grafo RDF(S) em XML.

13. Especificar Restrições com Quantificadores (usando guião)

Grave o anterior e regresse à ontologia da pizza.

a) Leia as ideias apresentadas na secção 4.8.1.

b) Construa as quantificações existenciais indicadas na secção 4.8.2.

c) Não prosseguir, por enquanto, para outra secção; i.e., ficar na página 48 e gravar o projecto.

14. Construir Indivíduos (usando guião)

Construa uma nova ontologia "`http://PAISES`".

Abra o "`01_ProtegeOWLTutorialP4_v1_3.pdf`" no capítulo 7 (pág. 89).

a) Tal como indicado na secção 7.1 construa o conceito `Country`.

b) Siga as indicações da secção 7.2 e construa a restrição `hasValue`. *Notar* que nesta versão, do Protégé, em vez de termos "*Superclass*" temos "*Subclass Of*".

c) Siga as indicações da secção 7.3 e construa uma "classe enumerada".

d) Não prosseguir, por enquanto, para outra secção; i.e., ficar na página 94 e gravar o projecto.

15. Construir Indivíduos (sem usar guião)

Grave o anterior e regresse à ontologia "`http://FAMILA`".

a) Definir `Pedro` e `Joana` como instância de `Pessoa`.

b) Definir `Joana` também como Feminino. *Atenção*: para que um indivíduo pertença a mais do que um conceito é preciso defini-lo, no separador "Individuals", num conceito e depois, selecionar o indivíduo, e em "Types" escolher "+" e, na janela que surge, no separador "Class hierarchy" selecionar o outro conceito.

c) Indicar que `Joana` tem como filho `Pedro`; selecionar `Joana` e em "Object property assertions" escolher "+", e na janela que surge escolher `temFilho` e `Pedro`.

d) Fazer inferência ("Reasoner \ Start reasoner ou Synchronize reasoner").

e) Notar o resultado da inferência sobre os indivíduos ("Members list (inferred")); o indivíduo `Joana` é classificado como `Mae`.

f) Grave e leia o ficheiro "`familia.owl`".

ADEETC - Mestrado em Engenharia Informática

Guia Aula Prática

Representação e Processamento de Conhecimento (RPC)

16. Um Problema para Modelar

Construa a ontologia "`http://SUPERMERCADO`".

Considere o "mundo" do supermercado (será o "Thing" e portanto não precisa de ser explicitamente representado). No supermercado existem Produtos e Etiquetas-de-Corredor. Vamos considerar os seguintes tipos de Produto: Peixe, Carne, Fruta, Água, Vinho, Gelado (disjuntos entre si). As Etiquetas-de-Corredor são as seguintes: Bebidas, Frescos, Congelados (disjuntos entre si).

Adicionalmente, existem outros conceitos que nos ajudam a classificar cada produto. Esses conceitos são: Habitat, Espécie, PeríodoValidade (disjuntos entre si). O Habitat pode ser Terrestre ou Aquático (disjuntos entre si). A Espécie pode ser Animal ou Vegetal (disjuntos entre si). O PeríodoValidade pode ser Curto, Médio, Longo (disjuntos entre si).

Existem também algumas relações to tipo objecto ("object property"): temHabitat, temEspécie, temPeriodoValidade, todas elas têm como domínio um Produto e como contradomínio o nome do conceito incluído no seu próprio nome. Existem ainda relações de tipo de dado ("datatype property"): temÁlcool, éDoce, temSabor, todas elas têm como domínio um Produto e como contradomínio um valor do tipo "booleano".

Sobre os produtos sabe-se que peixe é equivalente a "animal que vive na água" (i.e., `Produto and (temEspecie only Animal) and (temHabitat only Aquatico) and (temEspecie some owl:Thing) and (temHabitat some owl:Thing)` no Protégé) enquanto a carne é "animal terrestre". A fruta é um vegetal e a água não tem sabor, o vinho tem álcool e o gelado é doce.

Os produtos do corredor com etiqueta Bebidas caracterizam-se por: terem sabor ou terem álcool ou não terem sabor nem álcool. Os produtos de Frescos são: curto período de validade e têm origem vegetal ou têm origem animal e habitat terrestre ou aquático. Os Congelados têm período de validade médio ou longo.

a) Construa um modelo que descreva aquele "mundo".

b) Adicione uma "sardinha" como instância de Produto e diga que ela é "animal que vive na água". Faça a inferência "Reasoner \ Start reasoner" e verifique a inferência de "sardinha é um Peixe".

c) Construa 5 indivíduos e mostre a inferência para concluir "em que corredor está cada um deles".