

INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL

REPRESENTAÇÃO DO CONHECIMENTO E RACIOCÍNIO

ERICK GALANI MAZIERO



LISTA DE FIGURAS

Figura 3.1. Exemplo de um recorte de uma ontologia de mundo	12
Figura 3.2. Exemplo de rede semântica	14



LISTA DE TABELAS

Tabela 3.1. Tabela verdade com resultados da aplicação de operadores entre P e Q	7
--	---

EMANIP

SUMÁRIO

3 REPRESENTAÇÃO DO CONHECIMENTO E RACIOCÍNIO	5
3.1 Agentes baseados em conhecimento	5
3.1.1 Representação do Conhecimento	5
3.1.2 Lógica proposicional.....	6
3.1.3 Lógica de primeira ordem.....	8
3.1.4 Engenharia de Conhecimento	10
3.1.5 Engenharia ontológica.....	11
3.1.6 Redes semânticas.....	13
REFERÊNCIAS.....	15
GLOSSÁRIO	16

3 REPRESENTAÇÃO DO CONHECIMENTO E RACIOCÍNIO

3.1 Agentes baseados em conhecimento

Os problemas apresentados no capítulo anterior têm um conhecimento do problema de uma forma bastante limitada: conhecem o estado atual, os estados percorridos e os possíveis próximos estados, além de uma função que verifica se um estado é um estado objetivo ou não. Essa perspectiva não tem muitas chances de sucesso em problemas que admitem um grande número de estados, dadas as limitações computacionais, tanto de tempo de processamento quanto de memória, para a resolução do problema.

Na Inteligência Artificial, há o que se chama de **agentes baseados em conhecimento**, que, como o nome diz, utilizam uma **base de conhecimento** (KB - *Knowledge Base*). Um KB armazena **sentenças**, que, desaliado de um sentido puramente linguístico, representa uma **asserção** sobre o mundo. As sentenças são expressas em uma **linguagem de representação do conhecimento**. Uma sentença pode ser chamada de **axioma**, quando tal sentença é dada originalmente, isto é, sem derivação de outra sentença.

O conhecimento em um KB é geralmente dinâmico, ou seja, novas sentenças são adicionadas e outras podem ser removidas ou modificadas. Vamos referenciar a inserção de novas sentenças pela operação TELL e a busca por uma sentença, como ASK. Essas operações podem utilizar-se de **inferências** de novas sentenças a partir das existentes no KB.

3.1.1 Representação do Conhecimento

Mas, afinal, o que é conhecimento? Antes, considere como **dados** quaisquer sequências numéricas ou alfanuméricas que não possuem um significado associado (3, 4, 5, ...). Quando esses dados são organizados e rotulados (tamanho 3, camada 4, 5 unidades, ...), passam a ser chamados de **informação**, pois significam algo. **Conhecimento** é a composição e organização de informações que permitem

raciocínio por parte de um ser humano ou de uma máquina (objeto de tamanho 3, na camada 3, com 5 unidades, ...).

Tratando-se de representar o conhecimento de forma a permitir o raciocínio por parte de um agente inteligente, uma das formas mais antigas é o uso da **lógica**. Inclusive, diversos sistemas especialistas do início da IA tiveram a lógica como sua base de conhecimento e raciocínio.

Lógica é uma linguagem de propósito geral utilizada para descrever fatos (verdades) em um *mundo*. A lógica também possui mecanismos para operar sobre os fatos visando o raciocínio. Assim como outras linguagens, tem-se uma **sintaxe**, que define a formação das sentenças e uma **semântica**, que dá o significado às sentenças. Cada sentença é sempre avaliada em **verdade** ou **falso**.

Linguagens naturais, como o Português e o Inglês também são uma forma de representar conhecimento. Tanto é que o conhecimento deste capítulo está representado utilizando o Português. No entanto, como será percebido na continuação deste capítulo, a linguagem natural é muito complexa e tem muitos fenômenos de difícil tratamento computacional, como a ambiguidade. Portanto, vamos nos atentar a algumas formas simbólicas de representação do conhecimento cujo tratamento é mais plausível, computacionalmente.

3.1.2 Lógica proposicional

A **lógica proposicional** é um tipo de lógica bem simples. Sua sintaxe define **sentenças atômicas**, que consistem em um **símbolo proposicional**. Aqui definiremos que as palavras que iniciam com letra maiúscula (P, Q, Humano, Máquina, etc.) são símbolos proposicionais. Cada símbolo proposicional é sempre avaliado em **verdade** ou **falso**. **Sentenças complexas** são feitas a partir de sentenças mais simples com o uso de **conectivos lógicos**: \neg (*não*), \wedge (*e*), \vee (*ou*), \Rightarrow (*implica*) e \Leftrightarrow (*se e somente se*).

Por exemplo, o uso do operador de negação para o símbolo P é $\neg P$, que significa “não P” ou “P é falso”, ou “não é o caso de P”. Os outros operadores dependem de uma **tabela verdade**. Considere a tabela verdade ilustrada na Tabela

3.1, a seguir, com os operadores \neg (não), \wedge (e), \vee (ou), \Rightarrow (implica, condicional) e \Leftrightarrow (se e somente se, bicondicional).

Tabela 3.1. Tabela verdade com resultados da aplicação de operadores entre P e Q

P	Q	\wedge	\vee	\Rightarrow	\Leftrightarrow
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

Fonte: Google (2017)

Pela tabela verdade, a sentença $P \wedge Q$ (P e Q) assume valores falsos (F) quando pelo menos um dos símbolos preposicionais forem falsos e só assume valor verdadeiro (V), quando os dois símbolos são verdadeiros. O operador \vee (ou), no entanto, leva a um valor verdade quando pelo menos um dos símbolos é verdade e só assume falso quando os dois são falsos. Como exercício, interprete os outros dois operadores.

Embora o exposto acima pareça um tanto formal e teórico, suas aplicações são bem práticas na IA, principalmente em agentes baseados em conhecimento. Imagine um agente (um *Chatbot*, por exemplo) que deve responder “sim” (verdade) ou “não” (falso), para afirmações de um usuário. Ele deve avaliar perguntas como:

1. A terra é redonda?
2. A terra é quadrada **E** a lua é redonda?
3. A neve é branca **SE** a água é incolor?
4. A terra é quadrada **ou** a lua é redonda?
5. A terra **NÃO** é quadrada **E** a lua é redonda?

Baseado em nosso conhecimento de mundo, as sentenças 1, 3, 4 e 5 são verdadeiras (V) e a sentença 2 é falsa. Concorde?

Embora útil em alguns cenários, para representar um conjunto de sentenças (conhecimento), sua representatividade, em cenários mais complexos, é limitada. A lógica proposicional manipula bem conhecimentos que podem ser expressos pelos conectivos lógicos entre as sentenças atômicas. Mas e para expressar coisas como “Existe algum ... tal que...” ou “Para todo ...”. Como representar em lógica

proposicional as sentenças: “Ser mais jovem que outrem”, ou “Existe um X que é maior que todos os outros Xs”? Veja no próximo subtópico uma linguagem com maior poder de expressão.

3.1.3 Lógica de primeira ordem

A lógica de primeira ordem (LPO) tem poder de expressão maior que a lógica proposicional. Sendo estudada por décadas, a LPO é utilizada como base para outras linguagens como o Prolog. A LPO estende lógica proposicional. Enquanto esta se ocupa da existência dos fatos, aquela se ocupa da existência de objetos e de relações entre eles.

Na LPO, as **sentenças atômicas** têm o seguinte formato: $P(t_1, \dots, t_n)$. P é chamado de **predicado** e tem um ou n **argumentos**. A LPO também contém **quantificadores**, a saber, $\exists x$ (para todo x) e $\forall x$ (para qualquer x). Tem-se também os **termos**, que são constantes, e, geralmente, se referem a objetos do mundo real, concretos ou abstratos. Assim, podemos expressar fatos por meio de predicados, como homem(joão) ou mulher(maria). Esses predicados podem significar que joão é um homem e maria é uma mulher.

Para expressar os quantificadores, temos que definir o conceito de variável, que será denotada com o uso de uma letra latina minúscula, geralmente mais do final do alfabeto, tais como x, y, z e w . Essas variáveis podem ser pensadas como posições de memória que podem ser ocupadas ou instanciadas por valores como joão e maria. Desta forma, podemos ter:

- homem(x), em que x é um homem;
- mulher(x), em que x é uma mulher;
- pai(x, y), em que x é pai de y ;
- mãe(x, y), em que x é mãe de y .

E, dados os predicados anteriores, $\forall y$ (para qualquer y), se pai(x, y) então maisnovo(y, x). Em outras palavras, se x é pai de y , então y é mais novo que x .

Para um conhecimento como: *tudo filho é mais novo que seu pai*, definimos:

$$\forall y (pai(x, y) \rightarrow \exists x (maisnovo(y, x)))$$

Veja que o predicado *homem* admite apenas uma variável, sendo considerado um predicado unário. Já o predicado *maisnovo* é binário, pois admite duas variáveis.

Dada uma base de conhecimento expressa em LPO, algumas inferências podem ser feitas para responder a algumas questões. Por exemplo, imagine uma base de conhecimento em que se definem algumas relações familiares. Algumas conclusões podem ser inferidas.

Sejam:

- $pai(x, y)$: x é pai de y;
- $filho(x, y)$: x é filho de y;
- $irmão(x, y)$: x é irmão de y.

Se quisermos descobrir quem é *irmão* de uma dada pessoa, podemos codificar:

$$\forall x \forall y \forall z (pai(x, y) \wedge filho(z, x) \rightarrow irmão(y, z))$$

Se instanciarmos $y = pedro$, teremos que qualquer instância de z será *irmão* de *pedro*, veja:

$$\forall x \forall y \forall z (pai(x, pedro) \wedge filho(z, x) \rightarrow irmão(pedro, z))$$

Se nossa base de conhecimento contiver:

- $filho(carlos, amaral)$
- $filho(pedro, joão)$
- $filho(estância, amaral)$
- $pai(amaral, lorena)$
- $pai(amaral, pedro)$

Concluimos que *pedro* é irmão de *lorena*.

Para entendermos um pouco mais sobre os quantificadores, considere

$\forall x \exists y ama(x, y)$: Para todo x, existe um y, tal que x, ama y, em outras palavras, todo mundo ama alguém. Porém, se quisermos dizer que existe um alguém que é

amado por todo mundo, teremos que escrever $\exists y \forall x \text{ ama}(x, y)$: existe um y que, para todo x , x ama y .

Um agente baseado em conhecimento pode armazenar novas sentenças em BK, utilizando a operação TELL. Por exemplo, TELL(BK, pai(amaral, murilo)) adiciona em BK o conhecimento de que *amaral* é pai de *murilo*, gerando um novo irmão para *pedro*.

3.1.4 Engenharia de Conhecimento

Na seção anterior, foi definida a LPO como uma forma de representar um conhecimento (BK), com a finalidade de se raciocinar e resolver problemas. Para se gerar a BK, tem-se um processo chamado **engenharia de conhecimento**. Um engenheiro de conhecimento investiga o domínio do problema, aprende conceitos importantes e define uma representação formal dos objetos e relações entre esses objetos no domínio.

Nesse processo podem-se gerar bases de conhecimento para um domínio bem circunscrito ou abordagem de uma variedade maior de conhecimentos humanos. Quanto maior a variedade, maior a complexidade da BK e mais desafiadora é a engenharia de conhecimento.

As seguintes etapas são necessárias:

1. *Identificar a tarefa*: deve-se delinear as questões que a base de conhecimento suportará e os tipos de fatos disponíveis.
2. *Agregar conhecimento relevante*: ou o engenheiro deve ser especialista do domínio ou fazer uma aquisição de conhecimento com um especialista do domínio. Por exemplo, para gerar uma base de conhecimento sobre diagnósticos médicos de uma dada classe de doenças, um especialista nessa classe de doenças deve ser amplamente consultado.
3. *Definir um vocabulário de predicados e constantes*: aqui se convertem os conhecimentos adquiridos em predicados e constantes em LPO, por exemplo. Definem-se predicados unário, binários e suas significações, por exemplo pai(x , y) indica que x é pai de y , ou que y é pai de x ?

4. *Codificar o conhecimento geral do domínio*: o engenheiro de conhecimento escreve os axiomas (termos iniciais, não provindos de inferência) correspondente a todos os termos do vocabulário. Essa etapa permite ao especialista verificar se o conhecimento adquirido foi corretamente convertido na representação lógica. Quando erros são identificados, deve-se retornar ao passo 3.
5. *Codificar uma descrição da instância específica do problema*: envolve a escrita de sentenças atômicas sobre o problema, instanciando predicados com conhecimentos adquiridos.
6. *Formular consultas de inferência para obter respostas*: aqui pode-se testar o funcionamento do raciocínio sobre os axiomas e fatos codificados nas etapas anteriores, inferindo novos conhecimentos, conforme nossos objetivos de utilizar a base de conhecimento. Utiliza-se todo o poder de inferência da LPO, não necessitando que se escreva um programa para cada inferência que se deseja obter.
7. *Depurar a base de conhecimento*: Se um axioma estiver ausente da base, algumas consultas poderão não ser respondidas. Axiomas ausentes ou axiomas muito fracos podem ser identificados quando a inferência é interrompida de forma inesperada. Faz-se necessária a revisão dos passos anteriores, garantindo que as consultas desejadas sejam respondidas satisfatoriamente.

3.1.5 Engenharia ontológica

Nas seções anteriores o enfoque foi na forma de representar conhecimento, agora vamos tratar um pouco a respeito do conhecimento que deve ser armazenado em uma base de conhecimento.

A **engenharia ontológica** desempenha papel importante na representação do conhecimento, pois organiza tudo no mundo em uma hierarquia de categorias. Dessa forma, são definidas categorias de objetos, substâncias e medidas, eventos e conhecimentos abstratos.

A figura ilustra a criação de uma ontologia. No nível mais superior, encontra-se “QualquerCoisa”, indicando que essa ontologia pode organizar todos os conceitos do mundo real. Representar *tudo*, no entanto, é uma tarefa muito complexa e quase inatingível. Embora comece com “QualquerCoisa”, o que se especifica de cima para baixo na ontologia pode ser específico a um determinado domínio.

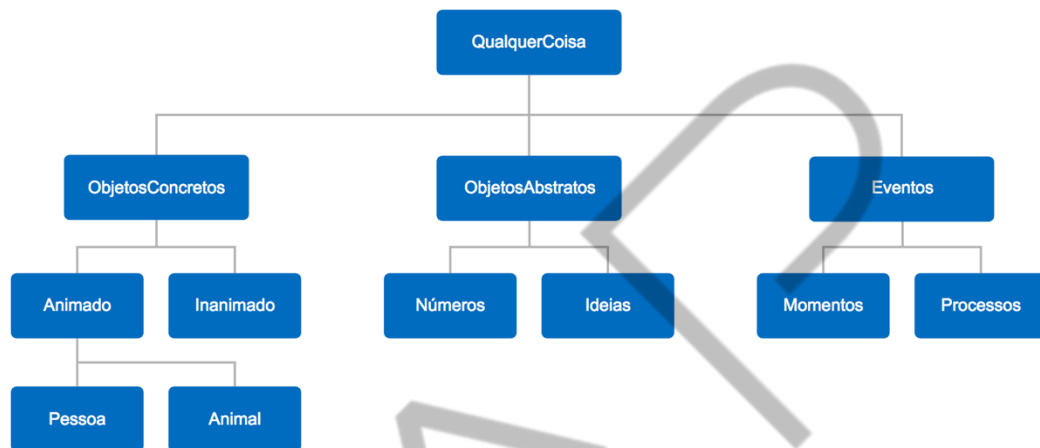


Figura 3.1. Exemplo de um recorte de uma ontologia de mundo
Fonte: Google (2017)

A Wikidata é uma iniciativa livre que busca organizar o conteúdo do mundo para leitura tanto humana quanto de máquinas. No momento da escrita desse material, mais de 87 milhões de fatos estão organizados hierarquicamente na Wikidata e abertos para uso público.

A ontologia ilustrada anteriormente é chamada de **ontologia superior**, pois define conceitos mais gerais na parte superior e vai especificando os conceitos à medida em que desce na ontologia.

A cada conceito que é especificado, deve-se levar em conta as similaridades dos objetos que pertencerão a cada ramo. Por exemplo, “Qualquer coisa” pode ser um “ObjetoConcreto” ou um “ObjetoAbstrato”. “ObjetoConcreto”, por sua vez, pode ser definido com o “Animado” (como pessoas e animais) ou “Inanimado” (objetos que não têm vida própria).

Embora, nesse material, tenha sido utilizada a LPO para representar os conhecimentos de mundo, as exceções (muito comuns no mundo real) não são facilmente tratadas. Por exemplo, dizer que *laranjas são da cor laranja* nem sempre é

verdade, pois laranjas podem ser verdes ou até amarelas, como boa parte das frutas e legumes.

A definição de **categorias** é parte importante da engenharia ontológica, pois embora a interação no mundo real ocorra com objetos específicos, boa parte do raciocínio feito sobre conceitos ocorre no nível de categorias. Interage-se com um carro específico, mas um consumidor, ao procurar um carro, busca na categoria de carros e não uma instância específica de carro.

Em LPO, representa-se uma categoria com predicados ou objetos. O predicado Carro(x), indica que x é um carro, ou um objeto Carro pode ser utilizado para representar todos os carros. Nesse tópico encontram-se muitas similaridades com o paradigma de Orientação a Objetos, muito comum na maior parte de linguagens de programação, com recursos como **herança de objetos**.

Outra definição importante a uma representação de conhecimento é a relação **parte de**. Objetos podem ser agrupados como **parte de** uma categoria de objetos. Por exemplo, temos ParteDe(roda, carro), ParteDe(motor, carro) ou ParteDe(freios, carro). Essa relação geralmente ocorre em **objetos compostos**, como o exemplo do Carro.

Na engenharia ontológica, outras definições devem ser levadas em consideração, como a definição de **medições, eventos, processos e intervalos de tempo**.

3.1.6 Redes semânticas

As **categorias** são os principais blocos de construção de bases de conhecimento em grande escala. As **redes semânticas** são um sistema que permite a visualização gráfica de uma base de conhecimento e algoritmos para a dedução de propriedades de um objeto.

Existem diversas variantes de redes semânticas, mas todas podem representar objetos individuais, categorias de objetos e relações entre os objetos. A notação gráfica típica representa os objetos e categorias nomeados em retângulos ou elipses e os conecta com arcos rotulados pelas relações entre os objetos ou categorias.

A figura ilustra uma rede semântica em que o “Carro A” é um objeto específico no mundo real, e é uma instância da categoria “Carro”, que é um tipo de “Veículo”. Por

exemplo, “Motocicleta” e “Caminhão” também poderiam ser tipos de “Veículo”. “Carro A” tem “Marca”, “Modelo” e “Proprietário”, cujos respectivos valores são “BMW”, “i8” e “Pedro”. Além disso, na rede semântica é representado um tipo de “Roda” (parte de um “Carro”), a “Roda de liga leve”.

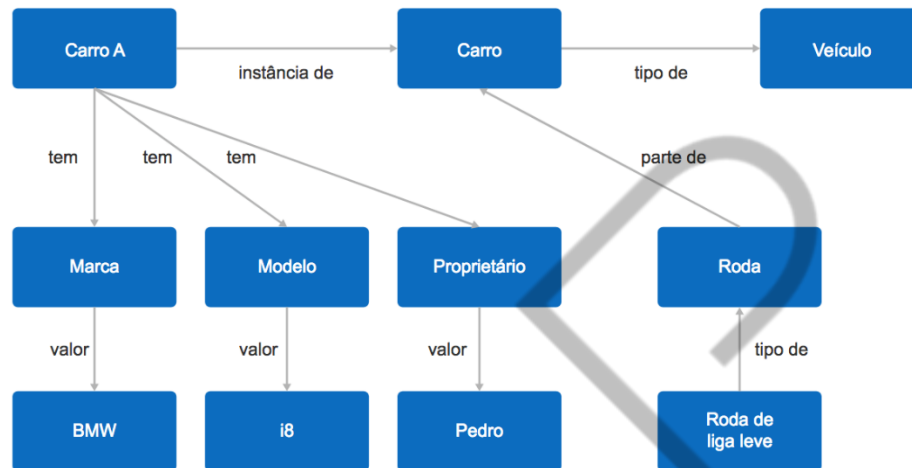


Figura 3.2. Exemplo de rede semântica
Fonte: Google (2017)

Como exercício, você poderia representar o conhecimento da rede semântica da figura em LPO.

REFERÊNCIAS

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2013.

PARDO, Thiago A. S.; REZENDE, Solange O. **Representação de Conhecimento**. 2015. 2018. Disponível em: <<http://wiki.icmc.usp.br/images/c/c3/Aula10-230t.pdf>>. Acesso em: 16 jan. 2018.

MOURA, Flávio L. C. **A Lógica de Primeira Ordem**. 2012. Disponível em: <<http://www.cic.unb.br/~flavio/disciplinas/graduacao/lc1/GramaticaLPredicados.pdf>>. Acesso em: 15 jan. 2018.

WIKIDATA. **Wikidata:Statistics**. [s.d.]. Disponível em: <<https://www.wikidata.org/wiki/Wikidata:Statistics>>. Acesso em: 25 jun. 2020.

GLOSSÁRIO

Chatbot	Programa que busca simular o comportamento humano em uma interação com um humano com o uso da linguagem natural.
Conectivos lógicos	Ou conectivos binários, que juntam duas sentenças, considerando os operandos de uma função.
Ontologia	Modelo de dados que representa um conjunto de conceitos em um domínio e os relacionamentos entre os conjuntos.
Orientação a Objetos	Modelo de análise, projeto e programação de software baseado na composição e interação entre as unidades do software, que representam objetos do mundo real.
Semântica	Estudo dos significados de objetos, palavras, etc., considerando o contexto em que ocorrem.