

Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso
Campus Cuiabá - “Octayde Jorge da Silva”

Lógica de Primeira Ordem

Vitor Bruno de Oliveira Barth

Cuiabá-MT
2018

Agenda

- Introdução à Lógica de Primeira Ordem
- Sintaxe, Semântica e Simbologia
- Engenharia de Conhecimento para Lógica de Primeira Ordem

Características da Lógica Proposicional

- É **declarativa**: sua sintaxe apresenta uma sentença cujo **valor verdade varia de acordo com o contexto**. Ou seja, **conhecimento e inferência são separados**.

Ex.: $1 + 1 = 10$ é FALSO em um sistema de numeração de base 10, mas é VERDADEIRO em um sistema de base 2.

- Consegue lidar com **informações parciais**.

Ex.: Como armazenar a expressão 'Existe um buraco em [2,2] OU [3,1]'?

Características da Lógica Proposicional

- **É composicional:** uma sentença é uma **função dos valores de suas partes**.

Ex.: $S_{1,4} \wedge S_{1,2}$ está relacionada com os significados de $S_{1,4}$ e $S_{1,2}$.

- **Não** permite **descrever um ambiente com muitos objetos** de maneira concisa.

Ex.: É necessário escrever uma regra específica sobre brisa e buracos para cada quadrado: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, $B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{2,3})...$

Representação da Lógica de Primeira Ordem

Assume que o **ambiente é composto** de

- **Objetos:** pessoas, casas, cores, teorias, jogos de futebol, guerras, anos...
- **Relações:** (*unárias*) vermelho, redondo, pontudo ...
(*n-árias*) irmão de, maior que, dentro de, ocorreu após, pertence à...
- **Funções:** pai de, amigo de, metade de, maior que, um a mais que...

Representação da Lógica de Primeira Ordem

Ex.: “um mais dois igual a três”

Objetos: um, dois, três, um mais dois.

Relação: igual

Função: mais.

‘Um mais dois’ é o objeto obtido como resultado ao aplicar a função ‘mais’ aos objetos ‘um’ e ‘dois’. ‘Três’ é outro nome para este mesmo objeto

Diferenças da Lógica de Primeira Ordem

- **Compromisso Ontológico:** o que se **assume** sobre a **realidade do ambiente**.
 - Lógica proposicional assume que uma **sentença é somente VERDADEIRA ou FALSA** em um **ambiente**.
 - Lógica de Primeira Ordem assume que o mundo é **composto de objetos**, e uma **sentença pode** ser VERDADEIRA ou FALSA de **acordo com a relação** entre os objetos.
 - Lógica Temporal assume que uma **sentença é VERDADEIRA ou FALSA em um tempo específico**.

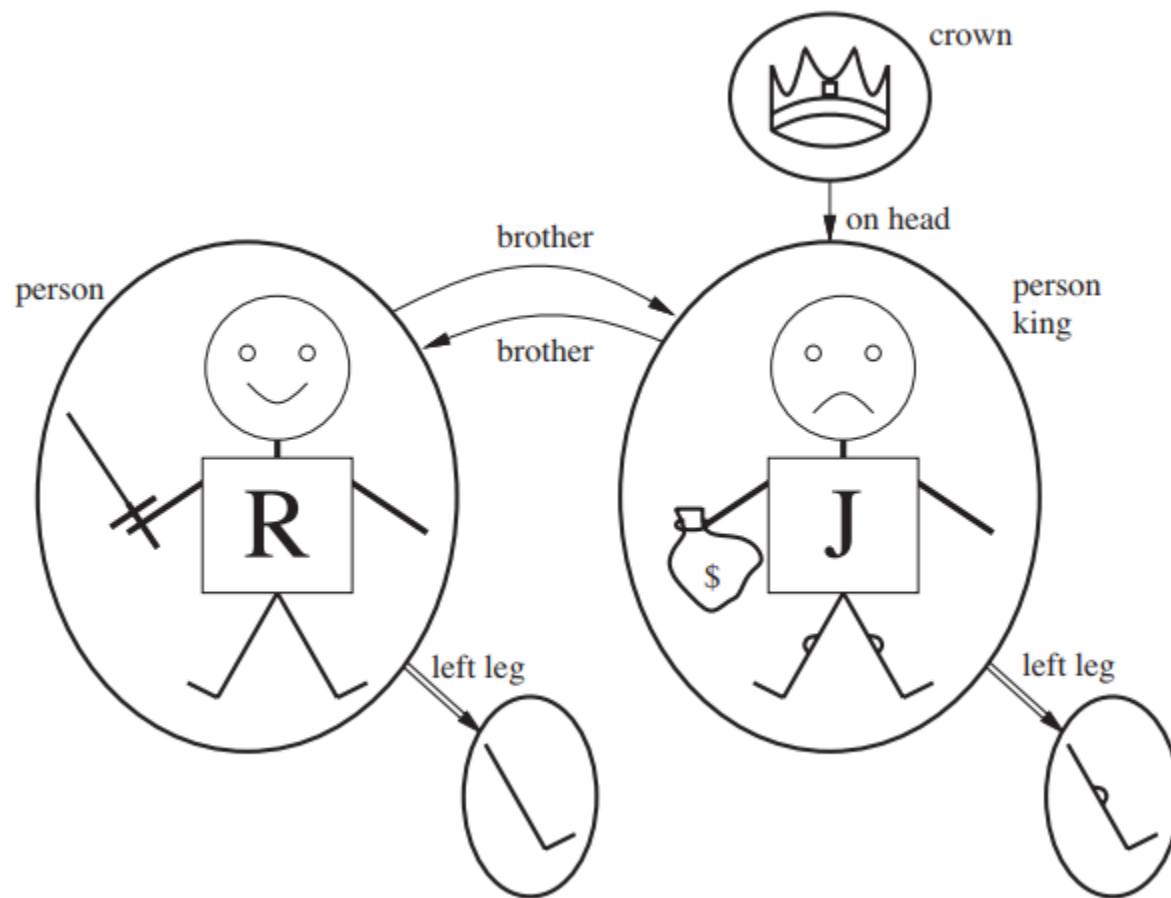
Diferenças da Lógica de Primeira Ordem

- **Compromisso Epistemológico:** o estado de conhecimento que uma sentença permite obter sobre um fato.
 - Lógica proposicional e Lógica de Primeira Ordem assumem que um fato pode ser **VERDADEIRO, FALSO** ou **não pode opinar**.
 - Lógicas Probabilísticas assumem que uma **sentença** possui um **grau de confiança**, que varia de 0 (total desconfiança) até 1 (total confiança).

Modelos para Lógica de Primeira Ordem

- Um modelo de Lógica de Primeira Ordem contém:
 - Um **conjunto não-nulo** de **objetos**.
 - **Objetos relacionados** de diversos modos.
 - Certas relações são chamadas de **funções**, e **devem ter** (teoricamente) **resposta para cada n-upla** de objetos de entrada.

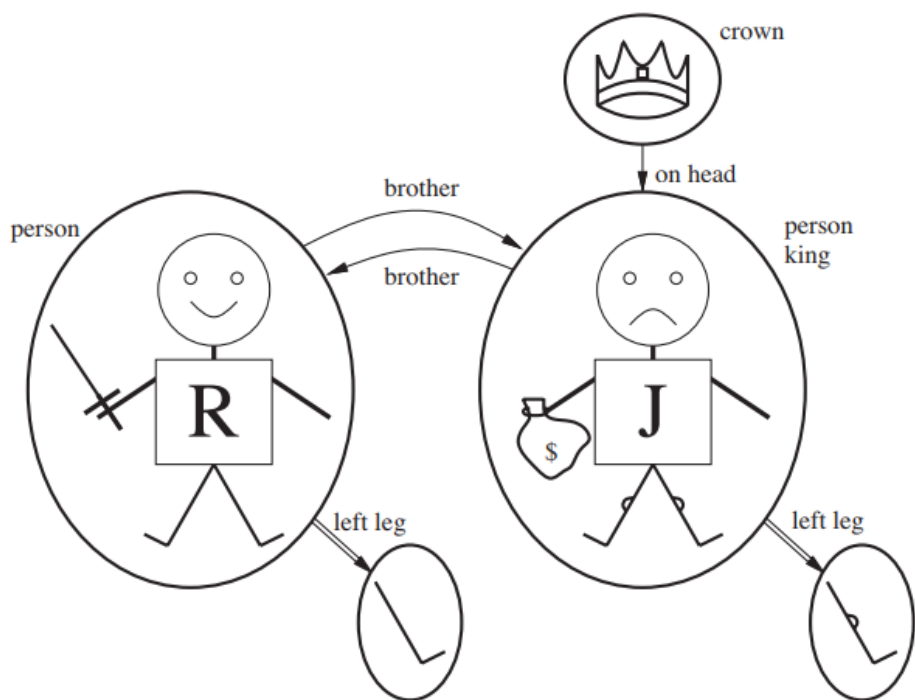
Exemplo de Modelo para FOL



Objetos:

- 1) Ricardo Coração de Leão, rei da Inglaterra entre 1189 e 1199.
- 2) O irmão mais novo, maldoso Rei João, que reinou entre 1199 e 1215
- 3) A perna esquerda de Ricardo
- 4) A perna esquerda de João
- 5) A coroa

Exemplo de Modelo para FOL



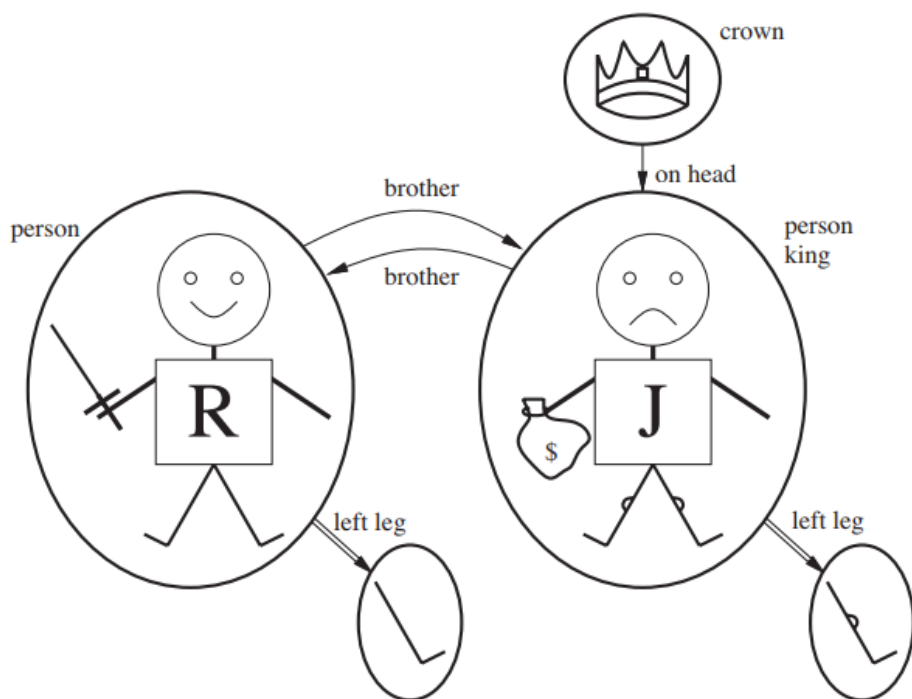
Relações entre Objetos: na figura, João e Ricardo são irmãos. De modo formal, a **relação** é um **conjunto de n-uplas contendo os objetos relacionados**. Deste modo, a relação de irmandade pode ser descrita por:

$\{ \langle \text{Ricardo Coração de Leão, Rei João} \rangle, \langle \text{Rei João, Ricardo Coração de Leão} \rangle \}$

A coroa está na cabeça do Rei João, então a relação 'na cabeça' é composta por somente uma n-upla: $\{ \langle \text{Coroa, Rei João} \rangle \}$

Exemplo de Modelo para FOL

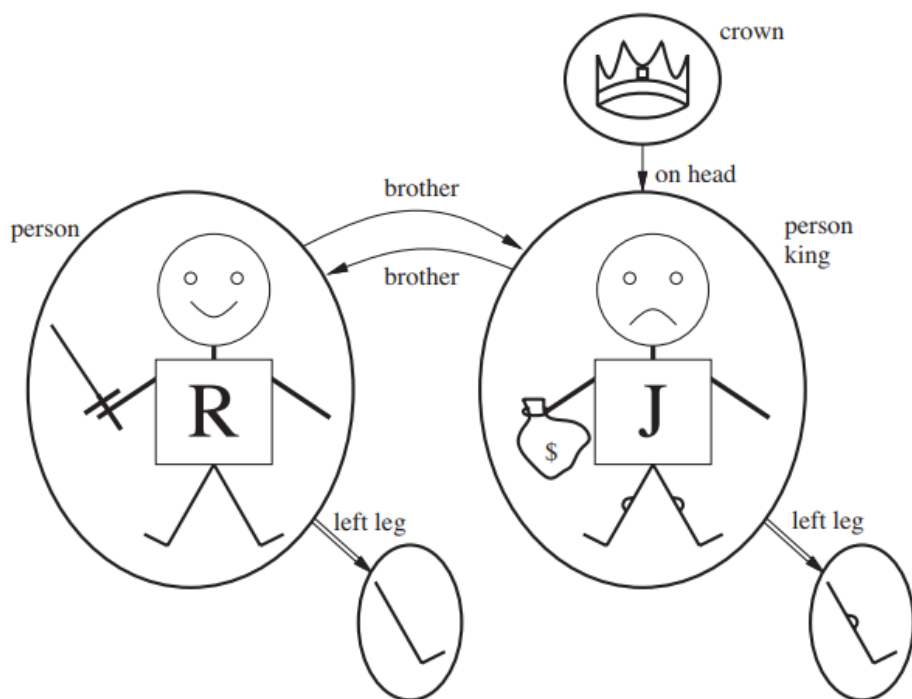
Relações entre Objetos: as relações ‘irmão’ e ‘na cabeça’ são **relações binárias** (ou seja, contém somente um par de objetos).



O modelo também contém relações unárias:

- ‘**pessoa**’ é propriedade de ambos João e Ricardo
- ‘**rei**’ é uma propriedade somente de João (até porquê Ricardo já está morto)
- ‘**coroa**’ é uma propriedade somente da coroa

Exemplo de Modelo para FOL



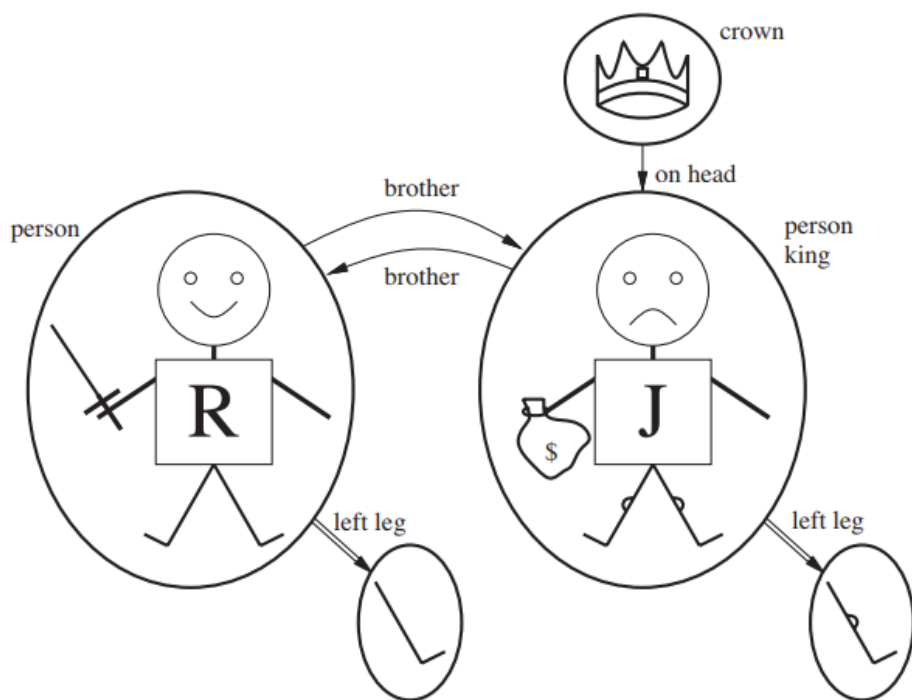
Alguns tipos de **relações** são **melhor descritas como funções**, de modo que um **dado objeto é relacionado a outro** objeto.

Por exemplo, toda pessoa do modelo têm uma perna esquerda. Deste modo, o modelo possui uma função unária 'perna esquerda' que é mapeada do seguinte modo:

«Ricardo Coração de Leão» → Perna esquerda de Ricardo.

«Rei João» → Perna esquerda de João.

Exemplo de Modelo para FOL



Tecnicamente, a função ‘perna esquerda’ **deve retornar um objeto para todo objeto**. A coroa deve ter uma perna esquerda, e a própria perna esquerda deve ter uma perna esquerda.

Felizmente, enquanto nenhuma sentença for criada envolvendo a perna esquerda de algo que não possui uma perna esquerda, essas technicalidades não são importantes.

Simbologia da Lógica de Primeira Ordem

Uma lógica de primeira ordem \mathcal{L} é construída sobre o seguinte conjunto de símbolos:

- Conectivos Proposicionais: \neg , \wedge , \vee e os atalhos \Leftrightarrow e \Rightarrow .
- Constantes Proposicionais: \top e \perp (VERDADEIRO e FALSO)
- Igualdade: $=$ (nem sempre incluso)
- Um conjunto de variáveis: $x_1, x_2, x_3, \dots, x_n$
- Quantificador Universal: \forall
- Quantificador Existencial: \exists
- Precedência: $\neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow$.

Simbologia da Lógica de Primeira Ordem

Termos

São **expressões que se referem a objetos**. Símbolos **constantes** são chamados de termos, mas **nem sempre é conveniente ter um símbolo para cada objeto**. Deste modo, podemos usar funções, como *PernaEsquerda*(João). Isto **não é** uma **sub-rotina** ou chamada de **método**, mas sim uma **referência atômica** à ‘perna esquerda de João’.

Simbologia da Lógica de Primeira Ordem

Sentenças Atômicas

São **expressões que se referem a relações**. É formado por um **predicado** seguido por **parênteses** e uma **lista de termos**. Exemplos:

a) *Irmão(João, Ricardo)*

b) *Casados(Mãe(João), Pai(Ricardo))*

Uma **sentença** é **VERDADEIRA** se a **relação** descrita por ela **existe entre** os **objetos** referidos como argumento.

Simbologia da Lógica de Primeira Ordem

Sentenças Complexas

Podem ser utilizados **conectivos** para se formar sentenças complexas.
Exemplos de Sentenças Complexas VERDADEIRAS:

a) $\neg \text{Irmão}(\text{PernaEsquerda}(\text{João}), \text{Ricardo})$

b) $\text{Rei}(\text{João}) \vee \text{Rei}(\text{Ricardo})$

c) $\neg \text{Rei}(\text{Ricardo}) \Rightarrow \text{Rei}(\text{João})$

Simbologia da Lógica de Primeira Ordem

Quantificadores

Ao formarmos uma lógica que permite a existência de objetos, é natural querer **expressar propriedades de um conjunto completo de objetos**.

Quantificador Universal (\forall)

Expressa regras como ‘todos os quadrados ao redor do Wumpus são fedidos’ e ‘todos os reis são pessoas’.

Ex.:

$$\forall x \text{ } Rei(x) \Rightarrow Pessoa(x) .$$

(Leia-se: ‘para todo x, se x é rei, então x é uma pessoa’)

Simbologia da Lógica de Primeira Ordem

Quantificador Existencial (\exists)

Expressa regras que dizem respeito à alguns dos objetos em um ambiente, sem nomeá-los.

Ex.:

$$\exists x \text{ Coroa}(x) \wedge \text{NaCabeça}(x, \text{João}) .$$

(Leia-se: ‘existe um x tal que x é uma coroa e está na cabeça de João’)

Simbologia da Lógica de Primeira Ordem

Igualdade

É VERDADEIRA se dois termos indicam o mesmo objeto.

Ex.: *Pai(João) = Henrique*

Também pode ser usada para indicar que dois objetos são diferentes:

$$\exists x,y \text{ Irmão}(x, \text{Ricardo}) \wedge \text{Irmão}(y, \text{Ricardo}) \wedge \neg(x = y)$$

A negação $\neg(x = y)$ evita que a sentença seja VERDADEIRA quando ambos x e y são atribuídos à João. Sendo assim, x e y devem ser estritamente objetos diferentes.

Semântica da Lógica de Primeira Ordem

Semântica de Banco de Dados

Imagine que queremos dizer que Ricardo possui dois irmãos somente: João e Jorge. Conseguimos capturar esse pensamento usando a sentença abaixo?

$$\text{Irmão}(\text{João}, \text{Ricardo}) \wedge \text{Irmão}(\text{Jorge}, \text{Ricardo})$$

Na verdade não: se Jorge e João se referirem ao mesmo objeto, então a sentença é verdadeira. É necessário indicar que $\text{João} \neq \text{Jorge}$. Além disso, essa sentença também é verdadeira em ambientes que Ricardo tem irmãos além de Jorge e João. Um exemplo de sentença completa, seria a abaixo:

$$\text{Irmão}(\text{João}, \text{Ricardo}) \wedge \text{Irmão}(\text{Jorge}, \text{Ricardo}) \wedge \text{João} \neq \text{Jorge}$$

$$\wedge \forall x \text{ Irmão}(x, \text{Ricardo}) \Rightarrow (x = \text{João} \vee x = \text{Jorge})$$

Semântica da Lógica de Primeira Ordem

Semântica de Banco de Dados

Para evitar problemas, são feitas algumas considerações:

a) *Unicidade de Nomes*: cada símbolo constante se refere à um objeto diferente.

b) *Ambiente Fechado*: toda sentença não VERDADEIRA é tida como FALSA.

c) *Compleitude de Domínio*: cada modelo não possui elementos além dos nomeados por símbolos constantes.

Uso da Lógica de Primeira Ordem

TELL e ASK

Assim como na Lógica de Primeira Ordem, podemos criar uma Base de Conhecimento KB usando as funções TELL e ASK.

Exemplo: TELL(KB, Rei(João))

TELL(KB, Pessoa(Ricardo))

E para verificar se uma sentença é verdadeira, usamos ASK:

ASK(KB, Rei(João))

Uso da Lógica de Primeira Ordem

TELL e ASK

Sentenças usando Quantificadores também são reconhecidas

Exemplo: ASK(KB, $\exists x$ Pessoa(x)).

Neste caso, a resposta é simplesmente VERDADEIRO, o que não é muito útil. Para obter todos objetos que atendem algum requisito, podemos usar ASKVARs

Exemplo: ASKVARs(KB, Pessoa(x))

Então, serão retornados dois objetos: {x | João} e {x | Ricardo}. Esse tipo de resposta chama-se **substituição** ou **lista de ligação**.

Engenharia de Conhecimento para FOL

Processo de Engenharia de Conhecimento

1. Identificar a Tarefa
2. Juntar o conhecimento relevante
3. Criar um vocabulário de predicados, funções e constantes
4. Codificar o conhecimento geral do domínio
5. Codificar uma descrição para uma instância do problema
6. Criar buscas (*queries*) que retornem informações sobre o estado
7. Realizar o *debug* da Base de Conhecimento

Obrigado!

Vitor Bruno de Oliveira Barth

vitor.barth@gmail.com

Agradecimentos



**INSTITUTO
FEDERAL**
Mato Grosso