Lógica de Predicados

Prof. Dr. Silvio do Lago Pereira

Departamento de Tecnologia da Informação

Faculdade de Tecnologia de São Paulo

Motivação

Há vários argumentos que não podem ser adequadamente formalizados e validados em lógica proposicional.

Exemplo

Sócrates é homem.

Todo homem é mortal.

Logo, Sócrates é mortal

- intuitivamente, podemos ver que este argumento é válido
- sua formalização em lógica proposicional resulta em $\{p, q\} \models r$
- porém, não há como mostrar que $\{p, q\} \models r$ é válido
- a validade deste argumento depende do significado da palavra “todo”
- para tratar este tipo de argumento precisamos da **lógica de predicados**

Linguagem formal: elementos básicos

A linguagem formal da lógica de predicados é **mais expressiva** que aquela da lógica proposicional.

Esta maior expressividade decorre do fato de as fórmulas da lógica de predicados serem compostas pelos seguintes elementos básicos:

- **objetos**
- **predicados**
- **conectivos**
- **variáveis**
- **quantificadores**

Linguagem formal: sintaxe

Objeto

é **qualquer coisa** a respeito da qual precisamos dizer algo

Na lógica de predicados, a noção de objeto é usada num sentido bastante amplo. Objetos podem ser:

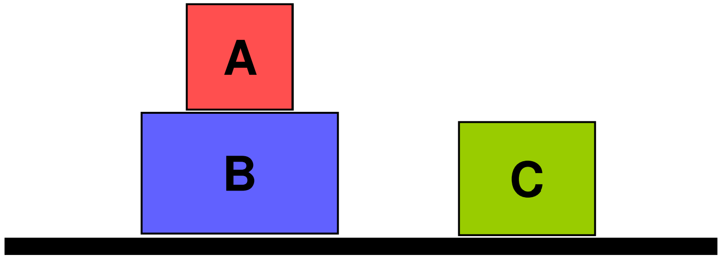
- **concretos**: a bíblia, a lua, ...
- **abstratos**: o conjunto vazio, a paz, ...
- **fictícios**: unicórnio, Saci-Pererê, ...
- **atômicos ou compostos**: um teclado é composto de teclas

Nomes de objetos devem iniciar com letra minúscula!

Linguagem formal: sintaxe

Predicado

denota uma **relação entre objetos** num determinado contexto



- `sobre(a,b)` : o bloco *A* está sobre o bloco *B*
- `cor(b,azul)`: o bloco *B* tem cor azul
- `maior(a,c)`: o bloco *A* é maior que o bloco *C*

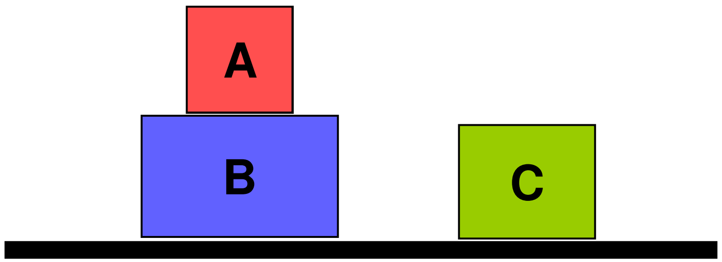
proposições atômicas!

Nomes de predicados também devem iniciar com letra minúscula!

Linguagem formal: sintaxe

Conectivo

forma **proposições compostas**, a partir de proposições atômicas



- `sobre(a,b)` ∧ `sobre(b,m)`: *A* está sobre *B* e *B* está sobre a mesa
- ¬ `cor(b,azul)`: a cor de *B* não é azul
- `maior(b,c)` ∨ `maior(c,b)`: o bloco *B* é maior que *C* ou *C* é maior que *B*

Linguagem formal: sintaxe

Variável

permite estabelecer fatos sobre objetos, sem nomeá-los explicitamente

- `bloco(X)` : *X* é um bloco
- `mesa(Y)` : *Y* é uma mesa
- `sobre(X,Y)` : *X* está sobre *Y*

não são proposições atômicas!

Note que proposições atômicas são sentenças que podem ter valor verdadeiro ou falso; mas não podemos dizer se `bloco(X)` é verdadeiro ou falso até que a variável `X` tenha sido **substituída** ou **quantificada**.

Nomes de variáveis devem iniciar com letra maiúscula!

Linguagem formal: sintaxe

Quantificador

permite estabelecer fatos sobre objetos, sem enumerá-los explicitamente

- Há dois quantificadores:

  Universal....: $\forall$X[bloco(X)] estabelece que todo objeto *X* é um bloco

  Existencial..: $\exists$Y[mesa(Y)] estabelece que algum objeto *Y* é uma mesa

- Estes quantificadores podem ser combinados numa mesma fórmula

  Todo bloco está sobre alguma coisa que é um bloco ou uma mesa

  $\forall$X[bloco(X) $\rightarrow$ $\exists$Y[sobre(X,Y) $\wedge$ (bloco(Y) $\vee$ mesa(Y))]]

Linguagem formal: semântica

Interpretação

- um conjunto não-vazio $\mathcal{D}$
- um mapeamento que associa cada objeto a um elemento fixo de $\mathcal{D}$
- um mapeamento que associa cada predicado a uma relação sobre $\mathcal{D}$

- **O quantificador universal denota conjunção**

  Por exemplo, para $\mathcal{D}$ = {a, b ,c, m}

  A fórmula `∀X[bloco(X)]` equivale a `bloco(a)` ∧ `bloco(b)` ∧ `bloco(c)` ∧ `bloco(m)`

- **O quantificador existencial denota disjunção**

  Por exemplo, para $\mathcal{D}$ = {a, b ,c, m}

  A fórmula `∃Y[mesa(Y)]` equivale a `mesa(a)` ∨ `mesa(b)` ∨ `mesa(c)` ∨ `mesa(m)`

- **Equivalências**

  $\neg\forall X[\alpha(X)] \equiv \exists X[\neg\alpha(X)]$

  $\neg\exists X[\alpha(X)] \equiv \forall X[\neg\alpha(X)]$

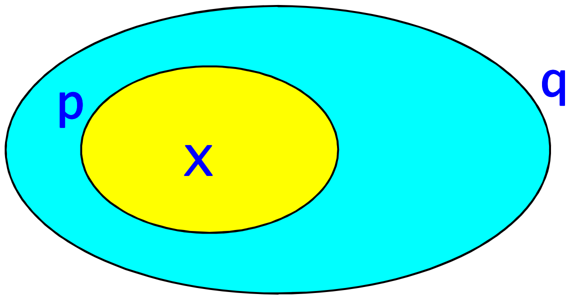
Representação de conhecimento

- Para facilitar a formalização se sentenças na lógica de predicados, destacamos quatro tipos de sentenças de especial interesse, denominadas **enunciados categóricos**:
  - **Universal afirmativo**: ***Todos*** *os homens são mortais*.
  - **Universal negativo**: ***Nenhum*** *homem é extra-terrestre*.
  - **Particular afirmativo**: ***Alguns*** *homens são cultos*.
  - **Particular negativo**: ***Alguns*** *homens* ***não*** *são cultos*.

Representação de conhecimento

Enunciado universal afirmativo

- é da forma $\forall x[p(X) \rightarrow q(X)]$
- estabelece que p é um subconjunto de q



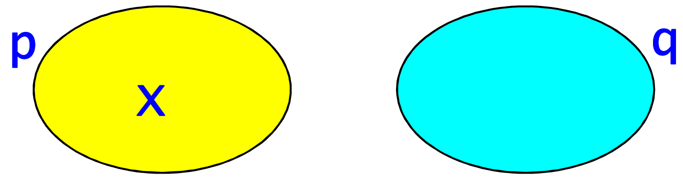
Exemplo:

- **Sentença....:** *Todos os homens são mortais*
- **Sintaxe.......:** $\forall x[h(X) \rightarrow m(X)]$
- **Semântica..:** para todo X, se $X \in h$ então $X \in m$

Representação de conhecimento

Enunciado universal negativo

- é da forma $\forall x[p(x) \rightarrow \neg q(x)]$
- estabelece que os conjuntos **p** e **q** são disjuntos



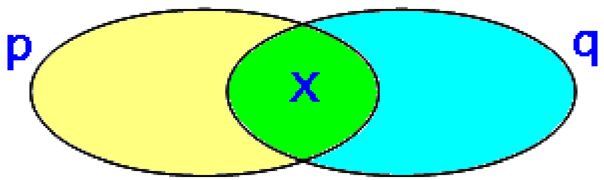
Exemplo:

- **Sentença....:** ***Nenhum homem é extra-terrestre***
- **Sintaxe.......:** $\forall x[h(x) \rightarrow \neg e(x)]$
- **Semântica..:** para todo x, se $x \in h$ então $x \notin e$

Representação de conhecimento

Enunciado particular afirmativo

- é da forma $\exists X[p(X) \land q(X)]$
- estabelece que os conjuntos **p** e **q** têm intersecção não-vazia



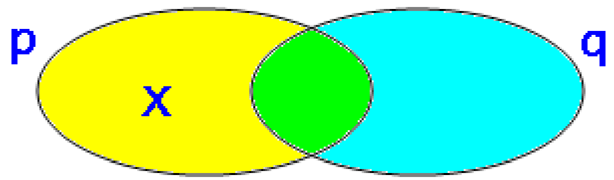
<u>**Exemplo**</u>:

- **Sentença....:** *Alguns homens são cultos*
- **Sintaxe.......:** $\exists X[h(X) \land c(X)]$
- **Semântica..:** existe X tal que $X \in h$ e $X \in c$

Representação de conhecimento

Enunciado particular negativo

- é da forma $\exists X[p(X) \land \neg q(X)]$
- estabelece que existem elementos em **p** que não estão em **q**



Exemplo:

- **Sentença....:** *Alguns homens não são cultos*
- **Sintaxe.......:** $\exists X[h(X) \land \neg c(X)]$
- **Semântica..:** existe X tal que $X \in h$ e $X \notin c$

Exercício 1. Formalize as sentenças a seguir usando lógica de predicados

- *Toda cobra é venenosa.*
- *Nenhuma bruxa é bela.*
- *Algumas plantas são carnívoras.*
- *Há aves que não voam.*
- *Tudo que sobe, desce.*
- *Existem políticos não são honestos.*
- *Não existe bêbado feliz.*
- *Pedras preciosas são caras.*
- *Ninguém gosta de impostos.*
- *Vegetarianos não gostam de açougueiros.*
- *Toda mãe ama seus filhos.*

Equivalência entre sentenças

- Há sentenças que podem ser escritas em mais de uma forma.
- Exemplo
  - Sentenças

    Nem tudo que brilha é ouro.

    Existe algo que brilha e não é ouro.
  - Fórmulas

    $\neg\forall x[b(x) \rightarrow o(x)]$

    $\exists x[b(x) \wedge \neg o(x)]$
  - Equivalência

    $\neg\forall x[b(x) \rightarrow o(x)]$

    $\equiv \neg\forall x[\neg b(x) \vee o(x)]$

    $\equiv \exists x \neg [\neg b(x) \vee o(x)]$

    $\equiv \exists x [b(x) \wedge \neg o(x)]$

Exercício 2. Verifique se os pares de sentenças são equivalentes

- *Nem toda estrada é perigosa.*
- *Algumas estradas não são perigosas.*

- *Nem todo bêbado é fumante.*
- *Alguns bêbados são fumantes.*

- *Nem todo ator americano é famoso.*
- *Alguns atores americanos não são famosos.*

Validação de argumentos

Exemplo

Sócrates é homem.

Todo homem é mortal.

Logo, Sócrates é mortal

- Formalização: $\{\, h(s), \forall X[h(X) \rightarrow m(X)] \,\} \models m(s)$
- Normalização: $\{\, h(s), \forall X[\neg h(X) \vee m(X)] \,\} \models m(s)$
- Refutação

(1) $h(s)$ Δ

(2) $\neg h(X) \vee m(X)$ Δ

----

(3) $\neg m(s)$ Hipótese

(4) $\neg h(s)$ RES(3,2) / {X=s}

(5) $\square$ RES(4,1)

instanciação de variável

Extração de respostas

Exemplo

Sócrates é homem.

Todo homem é mortal.

Consulta: ***Quem é mortal?***

- Formalização: $\{ h(s), \forall X[h(X) \rightarrow m(X)] \} \models \exists Y[m(Y)]$
- Normalização: $\{ h(s), \forall X[\neg h(X) \vee m(X)] \} \models \exists Y[m(Y)]$
- Refutação

```
(1)  h(s)             Δ
(2)  ¬h(X)∨m(X)       Δ
--------------------
(3)  ¬m(Y)            Hipótese
(4)  ¬h(Y)            RES(3,2) / {X=Y}
(5)  □                RES(4,1) / {Y=s}
```

$\neg \exists Y [m(Y)] \equiv \forall Y [\neg m(Y)]$

resposta da consulta

Instanciação de variáveis universais

Apenas variáveis universais podem ser corretamente instanciadas.

Variável universal: *“Todo cão é fiel a alguém”*

- **Fórmula......:** `∀X[cão(X) → ∃Y[fiel(X,Y)]]`
- **Instância.....:** `cão(rex) → ∃Y[fiel(rex,Y)] / {X=rex}`
- **Significado.:** *Se Rex é um cão, então Rex é fiel a alguém.*
- **Conclusão..:** a fórmula e sua instância têm significados coerentes

Variável existencial: *“Todo cão é fiel a alguém”*

- **Fórmula......:** `∀X[cão(X) → ∃Y[fiel(X,Y)]]`
- **Instância.....:** `∀X[cão(X) → fiel(X,ana)] / {Y=ana}`
- **Significado.:** *Todo cão é fiel a Ana.*
- **Conclusão..:** a fórmula e sua instância **não** têm significados coerentes

Skolemização de variáveis existenciais

Supomos a existência de uma *função* que dá o valor correto para a variável.

Variável existencial: *"Todo cão é fiel a alguém"*

- **Fórmula......:** $\forall X[\text{cão}(X) \rightarrow \exists Y[\text{fiel}(X,Y)]]$
- **Instância.....:** $\forall X[\text{cão}(X) \rightarrow \text{fiel}(X,\text{dono}(X))]$ / $\{Y=\text{dono}(X)\}$
- **Significado.:** *Todo cão é fiel a seu dono.*
- **Conclusão..:** a fórmula e sua instância têm significados coerentes

- A suposição destas funções foi originalmente proposta por *Thoralf Skolem*.
- A função deve ter como argumentos todas as variáveis que são globais a ela.
- Se não houver variáveis globais, em vez de função, podemos usar uma constante.
- Daqui em diante vamos considerar apenas variáveis universais.

Unificação

Unificação

é o processo de encontrar um conjunto minimal de substituições que torna duas fórmulas idênticas (a fim de que possamos usar resolução).

Algoritmo de unificação

Para unificar duas fórmulas atômicas (sem variáveis em comum):

- Compare as fórmulas até achar uma discrepância ou atingir o final de ambas.
- Ao encontrar uma discrepância:
  - Se nenhum dos elementos envolvidos for uma variável, finalize com ***fracasso***.
  - Caso contrário, substitua todas as ocorrências da variável pelo outro elemento e continue a comparação das fórmulas.
- Ao atingir o final de ambas as fórmulas atômicas, finalize com ***sucesso***.

Unificação

Prolog implementa unificação por meio do predicado predefinido =/2.

Exercício 3. Usando Prolog, verifique se os pares de fórmulas podem ser unificados

- `?- gosta(ana,X) = gosta(Y,Z).`
- `?- primo(X,Y) = prima(A,B).`
- `?- igual(X,X) = igual(bola,bala).`
- `?- ama(deus,Y) = ama(X,filho(X)).`
- `?- cor(sapato(X),branco) = cor(sapato(suspeito),Y).`
- `?- mora(X,casa(mãe(X))) = mora(joana,Y).`
- `?- p(X) = p(f(X)).`
- `?- p(f(Y),Y,X) = p(X,f(a),f(Z)).`

Fim