Programação Lógica

Aula 1

Alexsandro Santos Soares

prof.asoares@gmail.com

Aula 1

- Teoria
  - Introdução ao Prolog
  - Fatos, Regras e Consultas
  - Sintaxe do Prolog

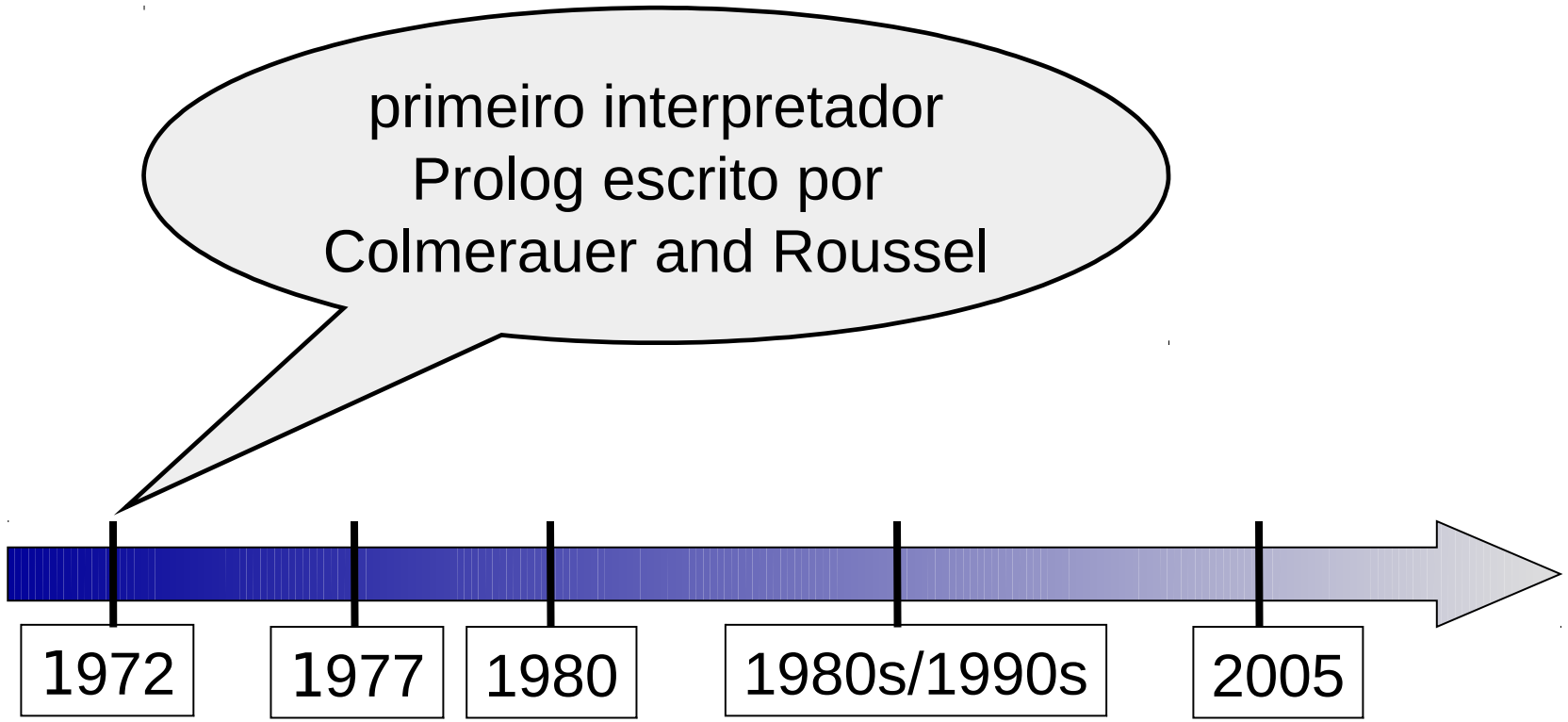
Objetivos desta aula

- Dar alguns exemplos simples de programas Prolog
- Discutir as três construções básicas do Prolog:
  - Fatos
  - Regras
  - Consultas
- Introduzir outros conceitos, tais como
  - o papel da lógica
  - unificação com o auxílio de variáveis
- Iniciar o estudo sistemático do Prolog via a definição de termos, átomos e variáveis

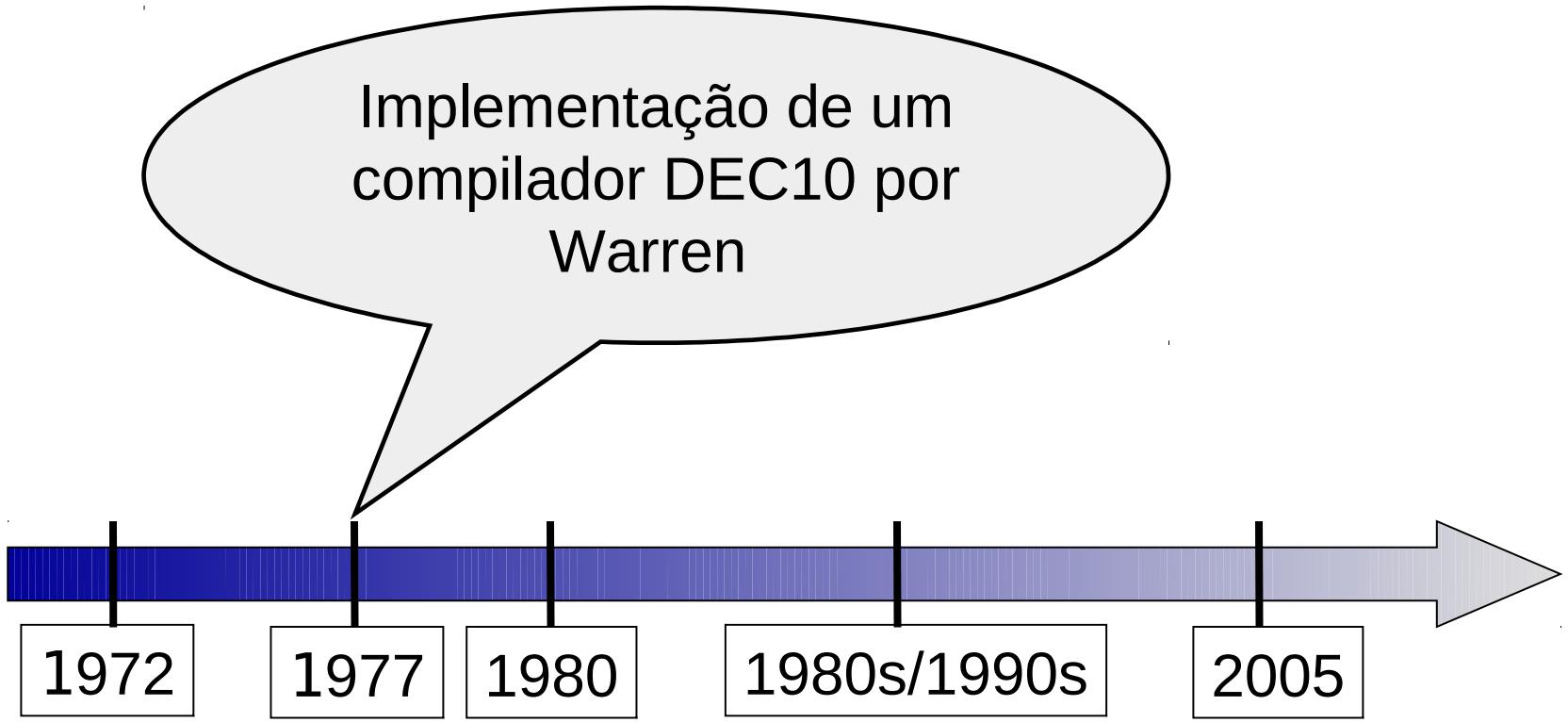
Prolog

- "Programação em Lógica"
- Declarativo
- Muito diferente de outras linguagens de programação (procedimentais)
- Bom para tarefas que exibem riqueza de conhecimento

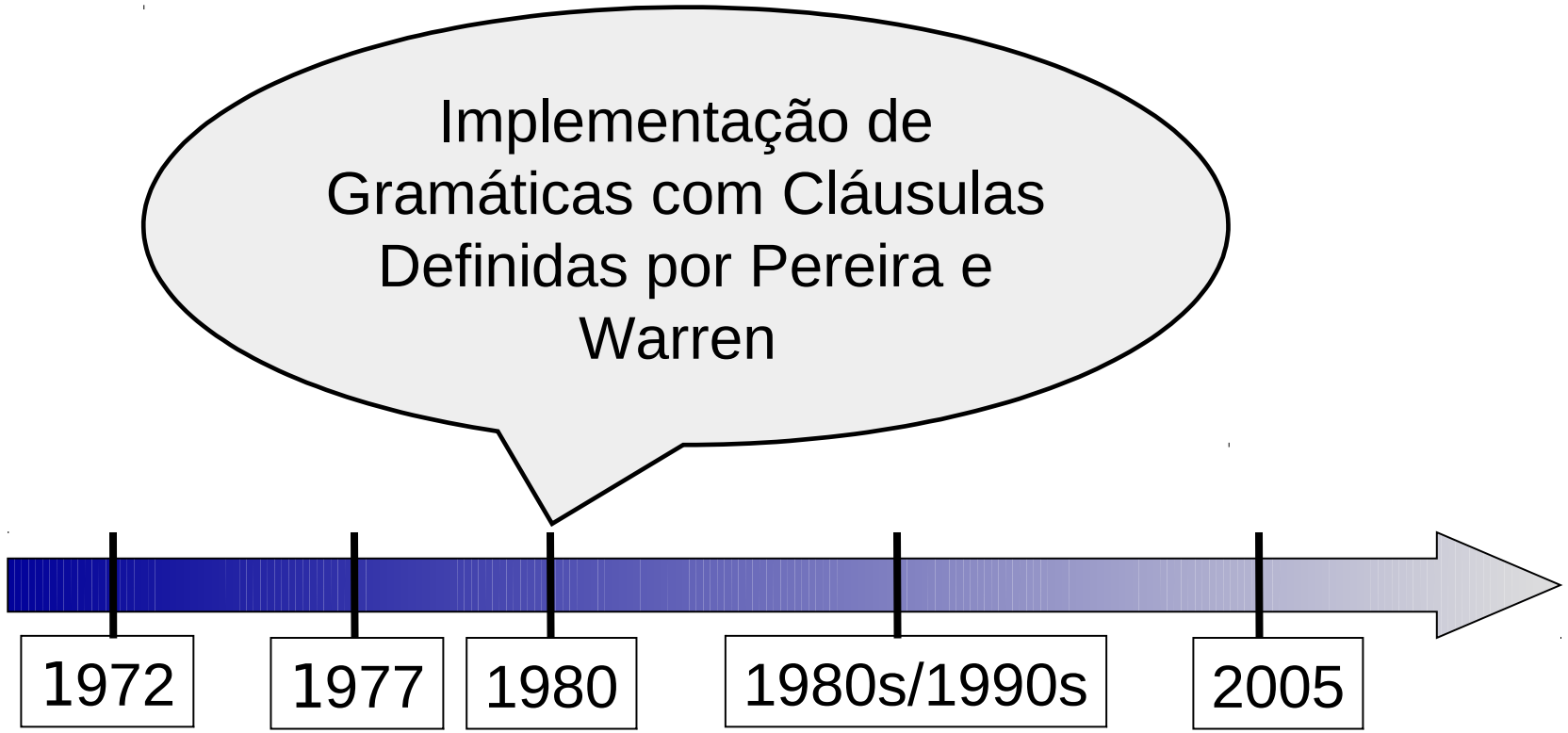
História do Prolog



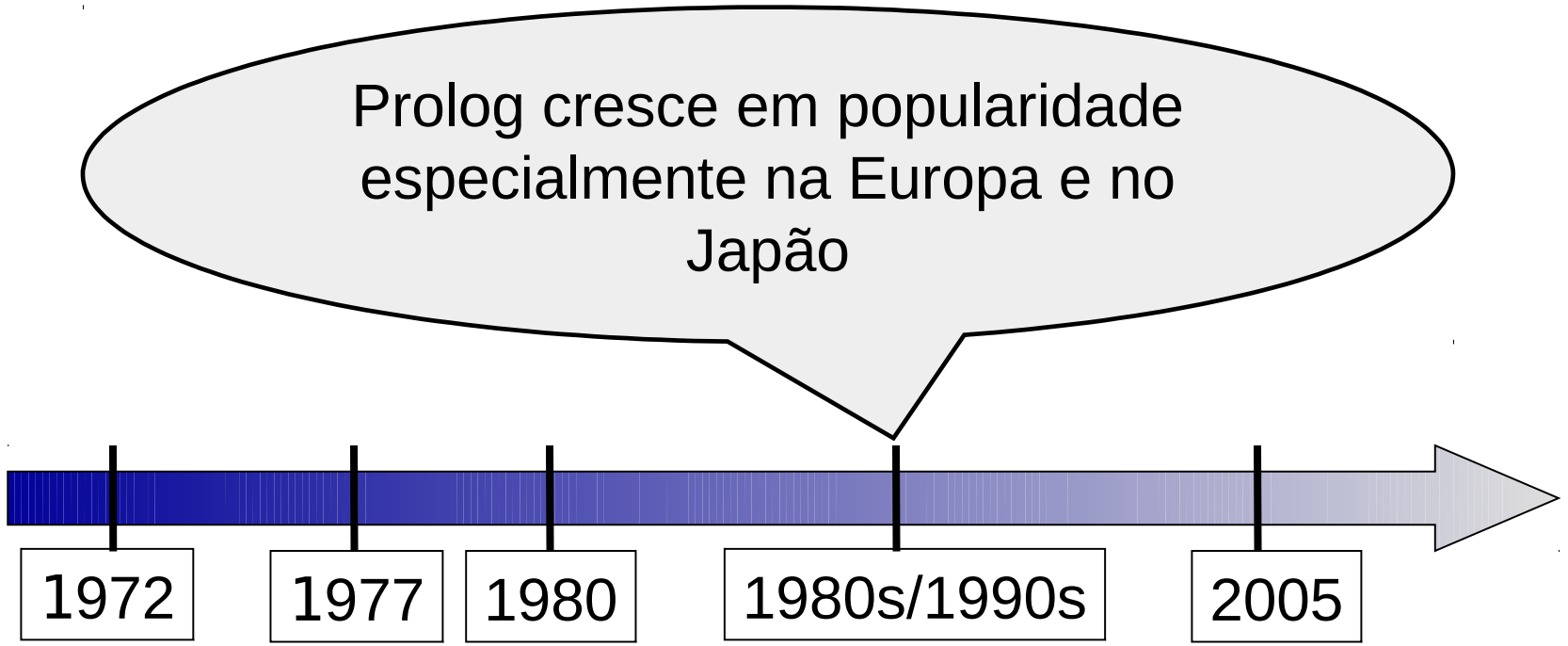
História do Prolog



História do Prolog

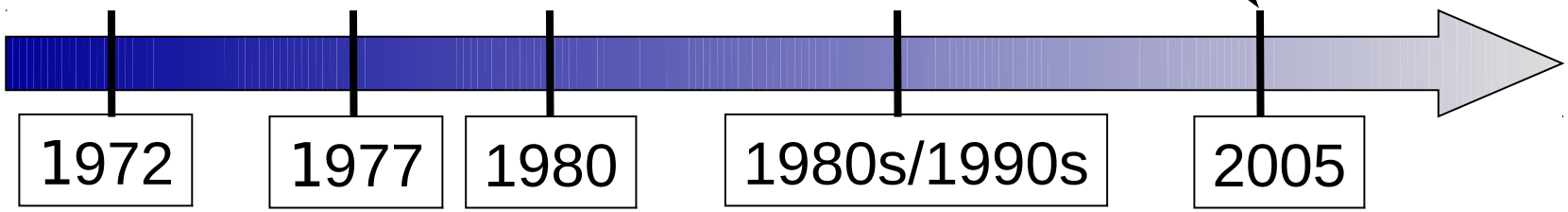


História do Prolog



História do Prolog

Prolog foi usado para programar a interface em língua natural na Estação Espacial Internacional pela NASA



Aplicações da Programação Lógica

- Sistemas Baseados em Conhecimento
  - sistemas que aplicam mecanismos automatizados de raciocínio para a representação e inferência de conhecimento
- Bancos de Dados “Inteligentes”
  - sistemas que empregam “agentes” de busca de dados com base em critérios
- Sistemas Especialistas
  - sistemas que emulam a especialização humana em algum domínio específico.
- Processamento da Linguagem Natural
  - usada para desenvolvimento de ferramentas para a comunicação homem-máquina em geral e para a construção de interfaces

Aplicações no mundo real

- Indústria de aviação, em soluções de planejamento e escalonamento:
  - PDC SCORE coordena 20% do tráfico aéreo do mundo
  - Sistemas usados em mais de 100 aeroportos
- Instituto Nacional de Meteorologia (INMET)
  - Previsão do tempo

Aplicações no mundo real

- Modelagem ambiental
  - Modelos matemáticos para a simulação do desenvolvimento de florestas
- Processamento de fala
  - Clarissa, desenvolvido pela NASA
- Telecomunicações
  - Sistema especialista da Ericssson para auxiliar o operador de redes
- Biotecnologia
  - Geração de ordens para instrumentos de análise de sequências de pyrosequenciamento.

Aplicações no mundo real

- Logística
  - Soluções ótimas em tempo-real para um fluxo contínuo de ordens de serviço
- Mineração de dados
  - Busca automática em bancos de dados por padrões e relacionamentos significantes
- Geração automática da primeira versão de documentos legais
  - DealBuilder

Aplicações no mundo real

- Análise de redes sociais e organizacionais
  - InFlow
- Suporte às decisões Clínicas
  - Arezzo
- Sistemas de gerenciamento de eventos
  - IBM Tivoli
- Construção de compilares
  - Erlang é uma linguagem criada pela Ericsson cuja primeira versão foi feita em Prolog.

Ideia básica do Prolog

- Descrever a situação de interesse
- Fazer uma pergunta
- Prolog deduz logicamente novos fatos sobre a situação que nós descrevemos
- Prolog retorna suas deduções como respostas

Consequências

- Pensar declarativamente, não procedimentalmente
  - Desafiador
  - Requer uma mentalidade diferente
- Linguagem de alto nível
  - Não tão eficiente quanto, digamos, C
  - Bom para prototipagem rápida
  - Útil em muitas aplicações de IA

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?-
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- mulher(maria).
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- mulher(maria).
true
?-
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- mulher(maria).
true
?- tocaGuitarra(joana).
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- mulher(maria).
true
?- tocaGuitarra(joana).
true
?-
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- mulher(maria).
true
?- tocaGuitarra(joana).
true
?- tocaGuitarra(maria).
false
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- tatuada(joana).
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- tatuada(joana).
false
?-
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- tatuada(joana).
ERROR: predicate tatuada/1 not defined.
?-
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- festa.
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- festa.
true
?-
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- concertoDeRock.
```

Base de Conhecimento 1

```
mulher(maria).
mulher(joana).
mulher(iolanda).
tocaGuitarra(joana).
festa.
```

```
?- concertoDeRock.
false
?-
```

Base de Conhecimento 2

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

Base de Conhecimento 2

feliz(iolanda). fato

escuta_musica(maria).

escuta_musica(iolanda):- feliz(iolanda).

tocaGuitarra(maria):- escuta_musica(maria).

tocaGuitarra(iolanda):- escuta_musica(iolanda).

Base de Conhecimento 2

feliz(iolanda). fato

escuta_musica(maria). fato

escuta_musica(iolanda):- feliz(iolanda).

tocaGuitarra(maria):- escuta_musica(maria).

tocaGuitarra(iolanda):- escuta_musica(iolanda).

Base de Conhecimento 2

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

fato

fato

regra

Base de Conhecimento 2

feliz(iolanda). fato

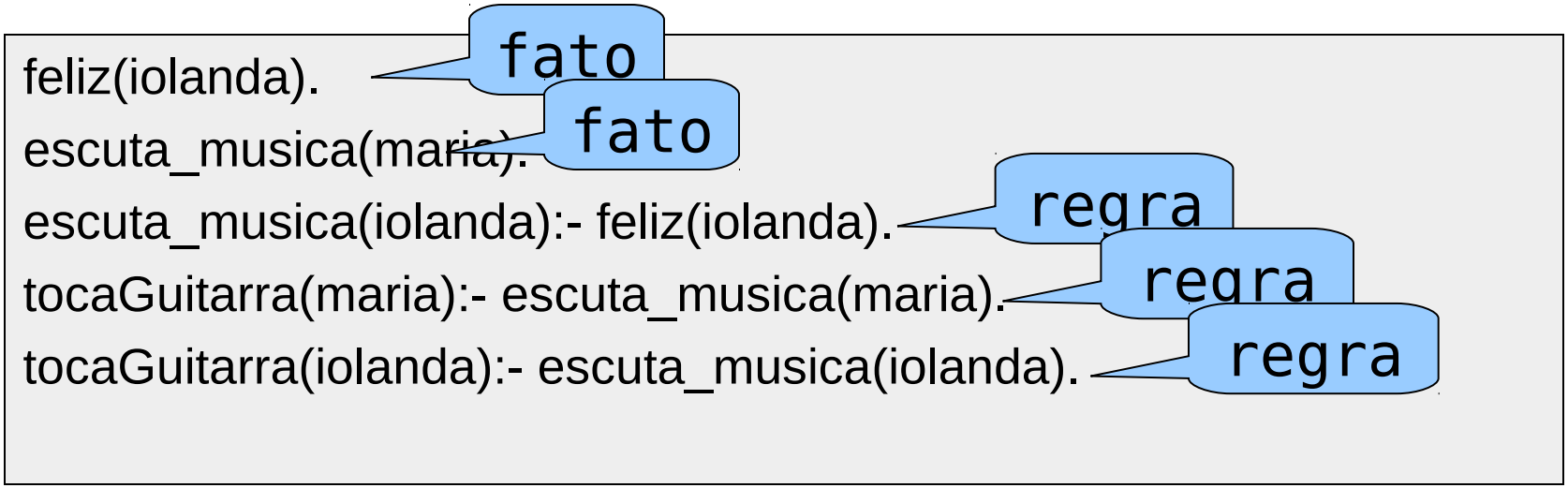
escuta_musica(maria). fato

escuta_musica(iolanda):- feliz(iolanda). regra

tocaGuitarra(maria):- escuta_musica(maria). regra

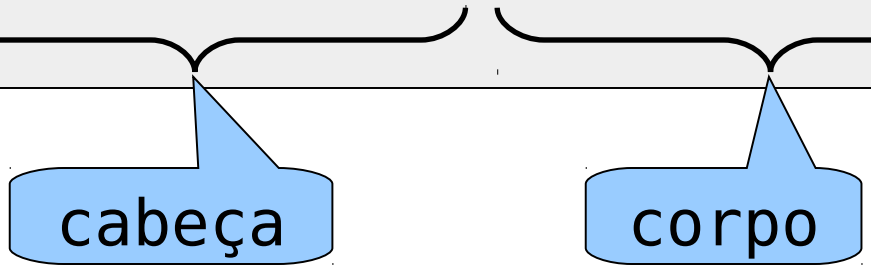
tocaGuitarra(iolanda):- escuta_musica(iolanda).

Base de Conhecimento 2



Base de Conhecimento 2

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```



Base de Conhecimento 2

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

```
?-
```

Base de Conhecimento 2

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

```
?- tocaGuitarra(maria).
true
?-
```

Base de Conhecimento 2

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

```
?- tocaGuitarra(maria).
true
?- tocaGuitarra(iolanda).
true
```

Cláusulas

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

*Existem cinco **cláusulas** nesta base de conhecimento:*

dois fatos e três regras.

O final de uma cláusula é marcado com um ponto final.

Predicados

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

*Existem três **predicados** nesta base de conhecimento:*

feliz, escuta_musica e tocaGuitarra

Base de Conhecimento 3

```
feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno).
tocaGuitarra(bruno):- escuta_musica(bruno).
```

Expressando Conjunção

```
feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno).
tocaGuitarra(bruno):- escuta_musica(bruno).
```

A vírgula “," expressa conjunção em Prolog

Base de Conhecimento 3

```
feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno).
tocaGuitarra(bruno):- escuta_musica(bruno).
```

```
?- tocaGuitarra(vicente).
false
?-
```

Base de Conhecimento 3

```
feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno).
tocaGuitarra(bruno):- escuta_musica(bruno).
```

```
?- tocaGuitarra(bruno).
true
?-
```

Expressando Disjunção

```
feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno).
tocaGuitarra(bruno):- escuta_musica(bruno).
```

```
feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno); escuta_musica(bruno).
```

Prolog e Lógica

- Obviamente Prolog tem algo a ver com lógica
- Operadores
  - Implicação **:-**
  - Conjunção **,**
  - Disjunção **;**
- Uso do *modus ponens*
- Negação

Base de Conhecimento 4

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

Variáveis em Prolog

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- mulher(X).
```

Instanciação de variáveis

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- mulher(X).
X=maria
```

Solicitando alternativas

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- mulher(X).
X=maria;
```

Solicitando alternativas

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- mulher(X).
X=maria;
X=joana
```

Solicitando alternativas

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- mulher(X).
X=maria;
X=joana;
X=iolanda
```

Solicitando alternativas

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- mulher(X).
X=maria;
X=joana;
X=iolanda;
false
```

Base de Conhecimento 4

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- ama(marcelo,X), mulher(X).
```

Base de Conhecimento 4

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- ama(marcelo,X), mulher(X).
X=maria
true
?-
```

Base de Conhecimento 4

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- ama(abobrinha,X), mulher(X).
```

Base de Conhecimento 4

```
mulher(maria).
mulher(joana).
mulher(iolanda).

ama(vicente, maria).
ama(marcelo, maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).
```

```
?- ama(abobrinha,X), mulher(X).
false
?-
```

Base de Conhecimento 5

```
ama(vicente,maria).
ama(marcelo,maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).

tem_ciumes(X,Y):- ama(X,Z), ama(Y,Z).
```

Base de Conhecimento 5

```
ama(vicente,maria).
ama(marcelo,maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).

tem_ciumes(X,Y):- ama(X,Z), ama(Y,Z).
```

```
?- tem_ciumes(marcelo,W).
```

Base de Conhecimento 5

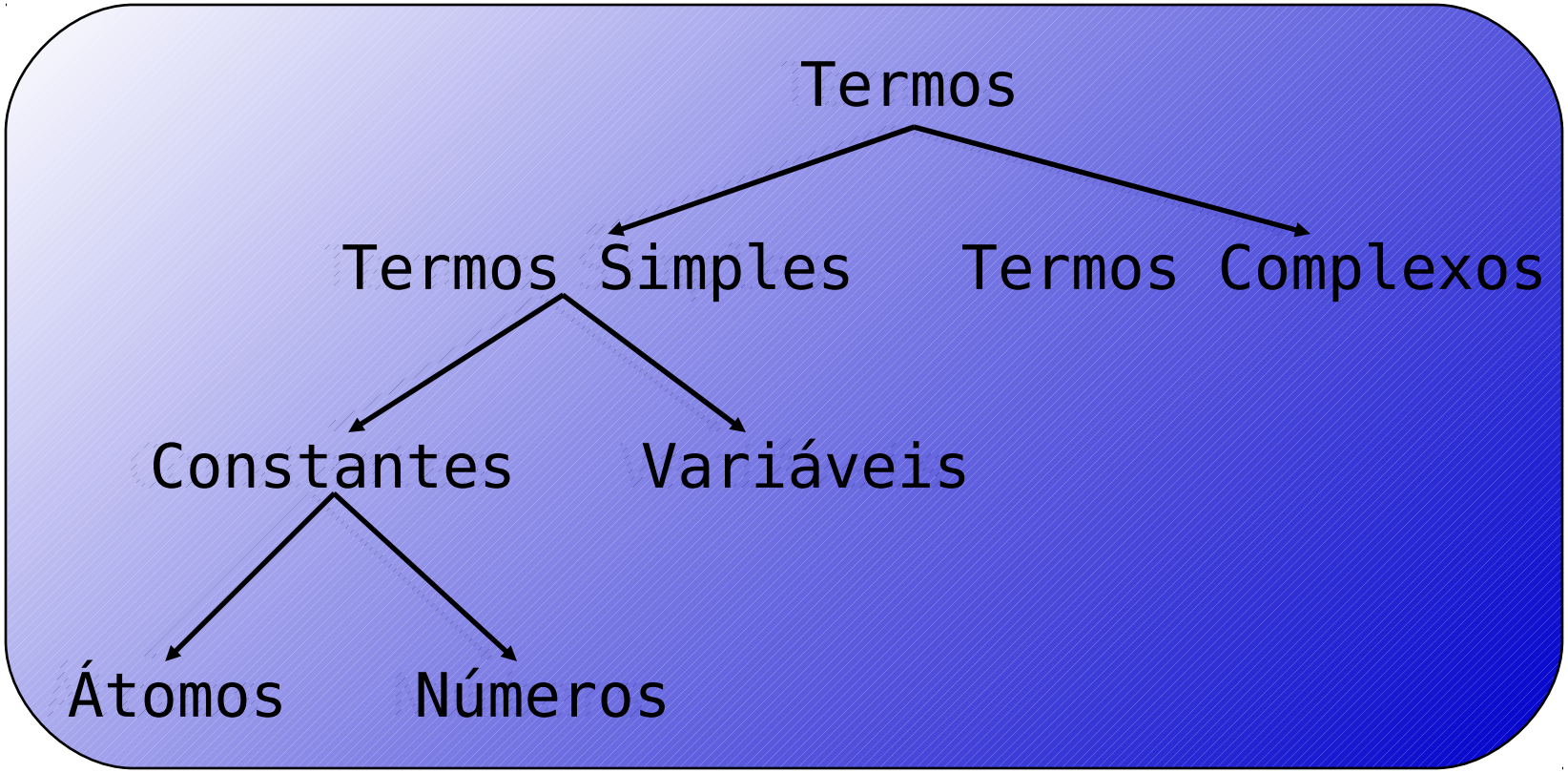
```
ama(vicente,maria).
ama(marcelo,maria).
ama(abobrinha, coelhinho).
ama(coelhinho, abobrinha).

tem_ciumes(X,Y):- ama(X,Z), ama(Y,Z).
```

```
?- tem_ciumes(marcelo,W).
W=vicente
?-
```

Sintaxe do Prolog

- Do que exatamente são construídos os fatos, regras e consultas?



Átomos

- É uma sequência de caracteres iniciando com uma letra minúscula e seguida de letras maiúsculas, minúsculas, dígitos ou sublinhado
  - *Exemplos*: **bruno**, **big_Mac**, **tocaGuitarra**

- Uma sequência arbitrária de caracteres entre apóstrofos
  - *Exs*: **'Vicente'**, **'Um e noventa e nove'**, **'@$%'**
- Uma sequência de caracteres especiais
  - *Exemplos*: **: , ; . :-**

Números

- Inteiros: 12, -34, 22342
- Ponto flutuante: 34573.3234

Variáveis

- Uma sequência iniciando com uma letra maiúscula ou sublinhado, seguida ou não de outras letras maiúsculas, minúsculas, dígitos ou sublinhados

- Exemplos:

  X, Y, Variavel, Vicente, _rotulo

Temos complexos

- Átomos, números e variáveis são os blocos construtores de termos complexos
- Termos complexos são construídos de um funtor diretamente seguidos por uma sequência de argumentos
- Os argumentos são colocados entre parênteses e separados por vírgulas
- O funtor deve ser um átomo

Exemplos de termos complexos

- Exemplos já vistos:
  - tocaGuitarra(joana)
  - ama(vicente, maria)
  - tem_ciumes(marcelo, W)
- Termos complexos dentro de termos complexos:
  - esconde(X,pai(pai(pai(bruno))))

Aridade

- O número de argumentos de um termo complexo é chamado de sua aridade

- Exemplos:

mulher**(maria)** é um termo com aridade 1
ama**(vicente,maria)** possui aridade 2
pai(pai(bruno)) possui aridade 1

Aridade é importante

- Em Prolog pode-se definir dois predicados com o mesmo funtor, mas com aridades diferentes
- Prolog trata isto como dois predicados diferentes
- Na documentação do Prolog a aridade de um predicado é normalmente indicada com o sufixo "/" seguido por um número para indicar a aridade

Exemplo de aridade

```
feliz(iolanda).
escuta_musica(maria).
escuta_musica(iolanda):- feliz(iolanda).
tocaGuitarra(maria):- escuta_musica(maria).
tocaGuitarra(iolanda):- escuta_musica(iolanda).
```

- Esta base de conhecimento define
  - feliz/1
  - escuta_musica/1
  - tocaGuitarra/1

Resumo desta aula

- Exemplos simples de programas Prolog
- Introdução de três construções básicas em Prolog:
  - Fatos
  - Regras
  - Consultas
- Discussão de outros conceitos, tais como
  - O papel da lógica
  - Unificação com o auxílio de variáveis
- Definição de construções do Prolog: termos, átomos e variáveis

Próxima aula

- Discussão de **unificação** em Prolog
- Estratégia de busca do Prolog