

Programação Lógica, 16-17

Inês Dutra
DCC-FCUP

ines@dcc.fc.up.pt (gabinete: 1.31)

September 21, 2016

Conteúdo

- Introdução às linguagens lógicas.
- Diferenças entre linguagens lógicas, funcionais e procedurais.
- Introdução à linguagem Prolog.
- Relação entre Prolog e a lógica.
- Relação entre Prolog e algoritmos de busca.
- Relação entre Prolog e bases de dados (Datalog).
- Sintaxe e semântica da linguagem Prolog.
- Tipos básicos de dados.
- Manipulação de listas. Recursão.
- Diferenças de listas.
- Boas práticas de programação em Prolog.
- Controle do espaço de busca. Operador de corte (cut).
- Depuração de programas Prolog.
- Satisfação de Restrições (Constraint Logic Programming)
- Implementação de Prolog (máquina WAM).
- Programação avançada em Prolog.

Bibliografia

- Leon Sterling and Ehud Shapiro, “The Art of Prolog”, MIT Press.
- Ivan Bratko, “Prolog Programming for Artificial Intelligence”, Addison-Wesley.
- Peter M. Kogge, “The Architecture of Symbolic Computers”, McGraw-Hill.
- William Clocksin and Chris Mellish, “Programming in Prolog”, Springer-Verlag.
- Hassan Aït-Kaci, “Warren’s Abstract Machine – A Tutorial Reconstruction”, MIT Press.
- Pascal van Hentenrick, “Constraint Satisfaction in Logic Programming”, MIT Press.

Bibliografia

- Peter van Roy, “1983-1993–The Wonder Years of Sequential Prolog Implementation”, JLP, 1994, V. 20, pp. 385–441.

Web site, Moodle etc

- <http://www.dcc.fc.up.pt/~ines/aulas/1617/PL/PL.html>
- Por favor use o endereço: `ines@dcc.fc.up.pt`
- Listas de exercício e mini-testes serão colocados no Moodle
- Atendimento: enviar email para combinar dia/horário

Motivação

- Linguagem de alto nível.
- Poder de expressão.
- Formal.
- Manipulação fácil de símbolos.
- Prova de Teoremas e Processamento de Linguagem Natural.

Prova de Teoremas ou Dedução Automática

- Representação do conhecimento de forma *declarativa*, em lógica matemática.
- Mecanismo de inferência (provador de teoremas) deduz soluções para os problemas.
- Exceto para classes de problemas muito restritas, o espaço de busca pode crescer exponencialmente.
- Requer estratégias de busca inteligentes.
- Pouco sucesso após muito trabalho nos anos 60.

O que faz programação em lógica ser diferente de prova de teoremas?

- Programação em Lógica é Programação!
- Programador se preocupa com a eficiência e praticidade (corre e termina em tempo polinomial?)
- Utilização de uma lógica restrita (Cláusulas de Horn).
- Mecanismo de inferência simples e eficiente.
- Algoritmo = Lógica + Controle.
- Controle pode ser ignorado quando “lemos” o programa de forma declarativa.

Onde Prolog é utilizado? (taken from stackoverflow)

- The first Erlang interpreter was developed in Prolog by Joe Armstrong
- Erlang is a programming language used to build massively scalable soft real-time systems with requirements on high availability. Some of its uses are in telecoms, banking, e-commerce, computer telephony and instant messaging. Erlang's runtime system has built-in support for concurrency, distribution and fault tolerance

Onde Prolog é utilizado? (taken from stackoverflow)

Companies using Erlang:

- **Amazon** uses Erlang to implement SimpleDB, providing database services as a part of the Amazon Elastic Compute Cloud (EC2).
- **Yahoo!** uses it in its social bookmarking service, Delicious, which has more than 5 million users and 150 million bookmarked URLs.
- **Facebook** uses Erlang to power the backend of its chat service, handling more than 100 million active users.
- **WhatsApp** uses Erlang to run messaging servers, achieving up to 2 million connected users per server.
- **T-Mobile** uses Erlang in its SMS and authentication systems.
- **Motorola** is using Erlang in call processing products in the public-safety industry.
- **Ericsson** uses Erlang in its support nodes, used in GPRS and 3G mobile networks worldwide.

Onde Prolog é utilizado? (taken from stackoverflow)

- IBM's Watson uses Prolog for the NLP-part.
- Prolog was also used by NASA to build a software named "clarissa", for the Intl Space Station. Clarissa is a voice user interface for browsing space station procedures.
- There's also PrologBeans, which you can use to build even a web app (integrated with other languages)
- SICStus Prolog (<http://sicstus.sics.se>) has been running systems that handle a third of all airline tickets, and helping railways to operate their trains better.
- DealBuilder - automatic construction of legal documents
- Arezzo - "Clinical decision support"
- InFlow - Social network analysis (looking for terrorists)

Conceitos Básicos em Lógica

- Computação: método de “raciocínio formal”.
- “objetos” do raciocínio: sentenças sobre o mundo (*fatos* ou *regras*).
- “Computação lógica”: usa o conjunto de sentenças para prever ou provar a validade ou falsidade de outras sentenças.

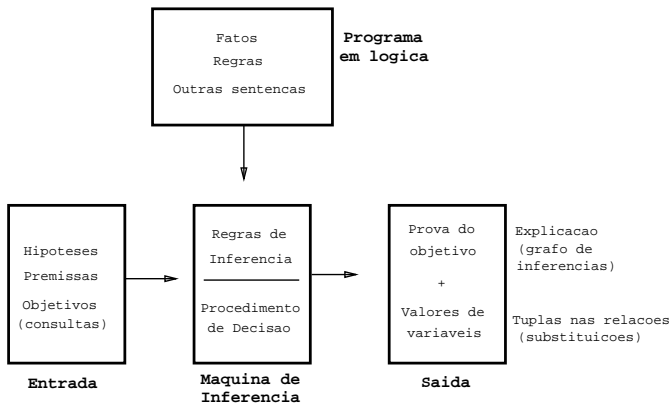
Conceitos Básicos em Lógica

- Modelo computacional básico: *máquina de inferência*.
- *Fatos*: entidades básicas em lógica, assume-se que são verdadeiros (axiomas).
- Exs: o preço deste livro é 49 euros, às 5 horas do dia 13/03/98 esteve chovendo, o fatorial de 3 é 6.
- Forma de expressar fatos: *relações*.
- Relação: conjunto de tuplas.
- Cada tupla: conjunto de objetos que compartilham as mesmas características ou possuem a mesma propriedade.
- Ex: relação `cor_do_cabelo(ines,cinzeno)`

Conceitos Básicos em Lógica

- Outras formas de relações: “Se A é verdade, então B é verdade”.
- Contradições não são permitidas: A é verdade e A não é verdade.
- Uma tupla não pode estar numa relação e ao mesmo tempo não estar.
- *Inferência*: conclui que uma sentença é verdadeira através da verificação de que outras sentenças são verdadeiras, sem ter que procurar exaustivamente pelo conjunto total de sentenças. Ex: Se x tem um filho/a, então x é pai.
Vítor é pai de Mariana.
- Não necessário procurar por todas as tuplas e-pai para provar que Vítor é pai.

Máquina de Inferência



Sistemas de Lógica Formal

- Sintaxe + Semântica.
- *Fórmulas bem formadas*: expressões sintaticamente corretas na linguagem.
- Conjunto de símbolos permitidos:
 - ▶ constantes;
 - ▶ funções;
 - ▶ predicados;
 - ▶ variáveis lógicas;
 - ▶ conectivos lógicos: implicação, conjunção, disjunção, negação, relacionais;
 - ▶ quantificadores.

Regra de Inferência

- Resolução generalizada com substituições
- Mesmas regras de cálculo de predicados proposicional podem ser aplicadas.
- Precisamos de mais regras para tratamento de variáveis.
- Substituição de variáveis por constantes individuais:
 $SUBST(\theta, \alpha)$.
- Ex: $SUBST(\{x/Sam, y/Pam\}, Likes(x, y)) = Likes(Sam, Pam)$

- três novas regras:

| | | |
|-------------------|---|--------------------|
| Elimin Universal | $\frac{\forall v \alpha}{SUBST(\{v/g\}, \alpha)}$ | Elimin Existencial |
| Intro Existencial | $\frac{\alpha}{\exists v SUBST(\{g/v\}, \alpha)}$ | |
- Importante: eliminação existencial deve fazer substituições por constantes que ainda **não** tenham aparecido no KB!