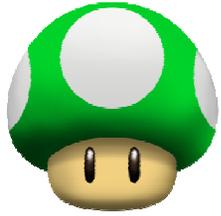


# INF 1771 – Inteligência Artificial

## Aula 10 – Prolog

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>



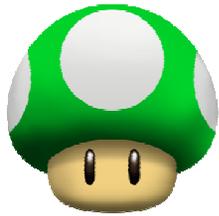
## Adicionando Novos Fatos a Base de Conhecimento

❏ O predicado **assert** é utilizado pelo Prolog para adicionar novas sentenças na base de conhecimento.

❏ **Exemplos:**

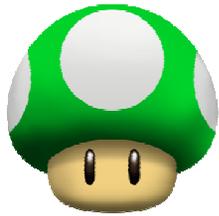
❏ `assert(homem(joao)).`

❏ `assert(filho(Y,X) :- progenitor(X,Y)).`



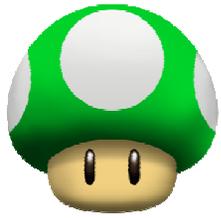
## Adicionando Novos Fatos a Base de Conhecimento

- ❏ O predicado **assert** permite adicionar **fatos** e **regras** a base de conhecimento.
- ❏ Normalmente, o SWI-Prolog compila o código de forma que **não é possível modificar** fatos durante a execução do programa.
- ❏ Para indicar ao Prolog que determinada sentença pode ser modificado durante a execução do programa é possível utilizar o predicado **dynamic**.
- ❏ **Exemplo:**
  - ❏ `:- dynamic homem/1.`



## Removendo Fatos da Base de Conhecimento

- ❏ Também é possível **remover** fatos e regras da base de conhecimento utilizando o predicado **retractall**.
- ❏ Funciona de forma similar ao assert.
- ❏ **Exemplos:**
  - ❏ `retract(homem(joao)).`
  - ❏ `retract(filho(Y,X) :- progenitor(X,Y)).`



# Predicados do SWI-Prolog

- ❏ O SWI-Prolog inclui diversas predefinidas para diversos usos, como por exemplo:
  - ❏ Manipulação de listas;
  - ❏ Comparação de tipos de dados;
  - ❏ Leitura e escrita de dados em arquivos;
  - ❏ Entrada e saída de dados pelo console;
  - ❏ Manipulação de arquivos;
  - ❏ Execução de comandos no sistema operacional;
  - ❏ Entre outros.
  
- ❏ <http://www.swi-prolog.org/pldoc/refman/>