

Lógica para Programação

LEIC-Tagus

P2

Prolog
Luísa Coheur

Programa

- Conceitos Básicos (Livro: 1.1)
- Lógica Proposicional (ou Cálculo de Predicados) — sistema dedutivo (2.1, 2.2.1, 2.2.2 e 2.2.4)
- Lógica Proposicional (ou Cálculo de Predicados) — resolução (3.1)
- Lógica de Primeira Ordem — sistema dedutivo (4.1, 4.2)
- Lógica de Primeira Ordem — resolução (5.1 e 5.2)
- Programação em Lógica (6)
- Prolog (7 + Apêndice A: manual de sobrevivência em Prolog)
- Lógica Proposicional (ou de Predicados) – sistema semântico (2.3, 2.4, 3.2)

Functores (7.17.4)

- Um termo composto corresponde à aplicação de uma letra de função (em prolog, designada por um functor) ao número apropriado de argumentos. Um functor é representado por um átomo.

functor(T, F, Ar): o termo T utiliza o functor F com aridade Ar

```
?- functor(maisAlto(hulk, capAmerica), maisAlto, 2).
```

```
true.
```

```
?- functor(maisAlto(hulk, xpto), Functor, Aridade).
```

```
Functor = maisAlto, Aridade = 2.
```

```
?- functor(T, xpto, 2).
```

```
T = xpto(_11766, _11768).
```

`arg(N, T, Arg)`: Arg é o N-ésimo argumento de T

?- `arg(1, maisAlto(hulk, capAmerica), hulk).`

`true.`

?- `arg(1, maisAlto(X, Y), hulk).`

`X = hulk.`

$T =.. L$: o primeiro elemento de L é o functor de T ; o resto de L são os argumentos de T

?- $T =.. [maisAlto, hulk, thor]$.

$T = maisAlto(hulk, thor)$.

?- $maisAlto(hulk, thor) =.. [Head \text{ --- } Tail]$.

$Head = maisAlto,$

$Tail = [hulk, thor]$.

call/1 tem sucesso se o objectivo que é o seu argumento tem sucesso

?- ..., L =.. [P, X, Y], call(L), ... % permite criar e executar o objectivo (run-time)

Nota: ?- ..., P (X, Y),... % dá erro