

# Answer Set Programming

## (*Programação com Conjunto de Resposta*)

### Parte – 01

Claudio Cesar de Sá<sup>1</sup>

Pesquisador Independente

## Roteiro

1. O que ASP?
2. Contexto Histórico
3. As diversas versões
4. Vamos ao Clingo nesta apresentação
5. Alguns elementos
6. Um exemplo: do Prolog ao ASP
7. Conclusões

# O que é o ASP?

- ▶ Uma linguagem de programação – Universidade de Potsdam (Alemanha)  
Universität Potsdam – 1999
- ▶ **Potassco**, the **Potsdam Answer Set Solving Collection** –  
<https://potassco.org/>
- ▶ <https://github.com/potassco-asp-course/> ... muitos slides
- ▶ Muitos slides, referências, códigos, vídeos no Youtube com os autores, etc
- ▶ Livros sobre ASP:

## Os livros:

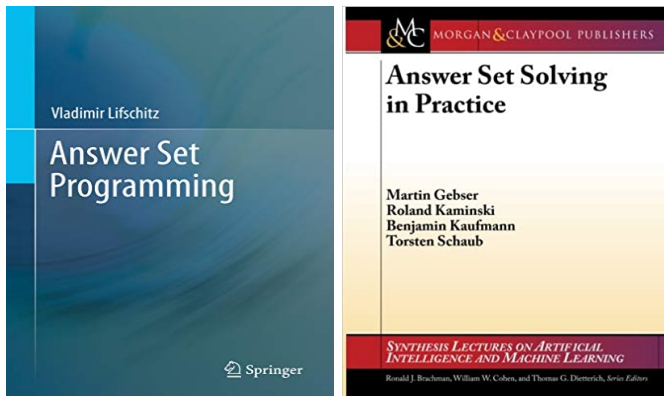


Figura: Estou usando o do Vladimir

# Características

- ▶ Mais declarativa que Prolog e seus predecessores (apenas a sintaxe lembra Prolog)
- ▶ Raízes em várias lógicas: LP, LPO, *default*, circunscrição
- ▶ Uso: problemas combinatoriais baseados em conhecimento declarativo
- ▶ Consistem de **decisões** e **restrições**
- ▶ Tudo isto na ordem de milhões!
- ▶ Na indústria: desde gerenciador de pacotes do Debian a sistemas da NASA

# Fundamentos – Fatos

```
1 p(1).  
2 p(2).  
3 %% LATER  
4 %% rule :- p(1) ; p(2).  
5 %% the constraint  
6 %% :- not p(2).
```

# Fundamentos – Fatos

```
1 p(1).  
2 p(2).  
3 %% LATER  
4 %% rule :- p(1) ; p(2).  
5 %% the constraint  
6 %% :- not p(2).
```

```
$ clingo ../01\_lesson.lp 0  
clingo version 5.4.0  
Reading from ../01\_lesson.lp  
Solving...
```

```
Answer: 1  
p(1) p(2)  
SATISFIABLE
```

```
Models          : 1  
Calls           : 1  
Time            : 0.001s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.  
CPU Time       : 0.001s
```

# Fundamentos – Conjunto de Fatos

```
1 %%clingo 02_lesson.lp 0
2
3 { p(aaa) ; p(bbb) }.
4 %*
5 {p(1..4)}.
6 %*
```



# Fundamentos – Conjunto de Fatos

```
1 %%clingo 02_lesson.lp 0
2
3 { p(aaa) ; p(bbb) }.
4 %*
5 {p(1..4)}.
6 %*
```

```
[ccs@vosges youtube_presentation_PORTUGUESE]$ clingo ../02_lesso
```

```
clingo version 5.4.0
```

```
Reading from ../02_lesson.lp
```

```
Solving...
```

```
Answer: 1
```

==> CONJUNTO VAZIO

```
Answer: 2
```

```
p(bbb)
```

```
Answer: 3
```

```
p(aaa)
```

```
Answer: 4
```

```
p(aaa) p(bbb)
```

# Fundamentos – Conjuntos e Restrições

```
1 %%clingo 03_lesson.lp 0
2
3 {p(1) ; p(2) ; p(3) ; p(4)}. %%% MESMO SET
4 3 {p(1 .. 4)} 3.
5
6 %% NAO PODE TER X MAIOR QUE 3
7 :- p(X), X > 3.
8 %:- p(X), X < 3.    ==> UNSATISFIABLE
9
10 %%% INCLUA p(2) nas respostas
11 %:- not p(2).
12
13 %%% print this contents -- true
14 %#show p/1.
```

# Fundamentos – Conjuntos e Restrições

```
1 %%clingo 03_lesson.lp 0
2
3 {p(1) ; p(2) ; p(3) ; p(4)}.   %%% MESMO SET
4 3 {p(1 .. 4)} 3.
5
6 %% NAO PODE TER X MAIOR QUE 3
7 :- p(X), X > 3.
8 %:- p(X), X < 3.   ==> UNSATISFIABLE
9
10 %%% INCLUA p(2) nas respostas
11 %:- not p(2).
12
13 %%% print this contents -- true
14 %#show p/1.
```

Answer: 1

p(1) p(2) p(3)

SATISFIABLE

Models : 1

Aqui são conjuntos .... flexíveis de se trabalhar!

# Fundamentos – Operador .. e o cartesiano

```
1 p(1..2, 1..8).  
2  
3 %%% As restricoes NAO podem  
4 %%% confrontar com que jah eh  
5 %%% uma VERDADE  
6 :- p(X,_) , X > 2.  
7 :- p(_ , Y) , Y > 8.
```

Saída:

# Fundamentos – Operador .. e o cartesiano

```
1 p(1..2, 1..8).  
2  
3 %%% As restricoes NAO podem  
4 %%% confrontar com que jah eh  
5 %%% uma VERDADE  
6 :- p(X,_) , X > 2.  
7 :- p(_ , Y) , Y > 8.
```

Saída:

```
$ clingo ../11_lesson.lp 0
```

Answer: 1

```
p(1,1) p(1,2) p(1,3) p(1,4) p(1,5) p(1,6)  
p(1,7) p(1,8) p(2,1) p(2,2) p(2,3) p(2,4)  
p(2,5) p(2,6) p(2,7) p(2,8)  
SATISFIABLE
```

Models : 1

Aqui **não** são mais conjuntos ....

# Construindo conjuntos

```
1 p( aaa ; bbb ; ccc ).
2 q( 777 ).
3
4 %%% p(X) eh LOCAL e q eh LIVRE
5 2 {r(X,Y) : p(X)} 7 :- q(Y).
6 %% de 2 a 7 conjuntos por resposta
7 %% O CONTRARIO
8 %1 {r(X,Y) : q(Y)} 1 :- p(X).
9 % OUTRA NOTACAO neste caso 1...1
10 %% {r(X,Y) : q(Y)} = 1 :- p(X).
11
12 % :- {r(X,Y) : p(X) , q(Y)} = 1 .
13 #show r/2.
```

Saída:

# Construindo conjuntos

```
1 p( aaa ; bbb ; ccc ).
2 q( 777 ).
3
4 %%% p(X) eh LOCAL e q eh LIVRE
5 2 {r(X,Y) : p(X)} 7 :- q(Y).
6 %% de 2 a 7 conjuntos por resposta
7 %% 0 CONTRARIO
8 %1 {r(X,Y) : q(Y)} 1 :- p(X).
9 % OUTRA NOTACAO neste caso 1...1
10 %% {r(X,Y) : q(Y)} = 1 :- p(X).
11
12 % :- {r(X,Y) : p(X) , q(Y)} = 1 .
13 #show r/2.
```

Saída:

```
$ clingo ../09_lesson.lp 0
```

```
Answer: 1
```

```
r(bbb,777) r(ccc,777)
```

```
Answer: 2
```

```
r(aaa,777) r(bbb,777) r(ccc,777)
```

```
Answer: 3
```

```
r(aaa,777) r(ccc,777)
```

```
Answer: 4
```

# Problema dos maridos, esposas e profissões

XXXXXXX



# Variável de Decisão e Uso

# Matriz de Decisão e Uso

# Definindo um Caminho

## Formalizando os Elementos

Estratégia deste Problema: **Fluxo** ( $\Phi$ )

# Equações Finais

## Próximos Passos:

- ▶ Implementar na OR-TOOLS com Python (próximo vídeo)
- ▶ Ferramenta livre mantida pela Google
- ▶ Suporta várias linguagens de *front-end*: C++, C#, Java e Python
- ▶ Vários *solvers*
- ▶ Vamos usar o *solver*: CP-SAT
- ▶ CP: *Constraint Programming*
- ▶ *Obrigado!*

## Contato e Comentários:

- ▶ <https://claudiocesar.wordpress.com/>
- ▶ <https://github.com/claudiosa>
- ▶ Email: [claudio.sa@udesc.br](mailto:claudio.sa@udesc.br)
- ▶ Email: [ccs1664@gmail.com](mailto:ccs1664@gmail.com)
- ▶ *Thank you so much!*