

# Answer Set Programming

(*Programação ... Conjunto de Respostas*)

## Um Panorama

Claudio Cesar de Sá<sup>1</sup>

Pesquisador Independente

## Roteiro

1. O que ASP?
2. Histórico
3. Sistemas relacionados a ASP
4. Esta apresentação
5. Alguns elementos
6. Um exemplo: do Prolog ao ASP
7. Conclusões

## Roteiro

1. O que ASP?
2. Histórico
3. Sistemas relacionados a ASP
4. Esta apresentação
5. Alguns elementos
6. Um exemplo: do Prolog ao ASP
7. Conclusões

**Em resumo: uma apresentação de algumas idéias elementares de ASP!**

# O que é o ASP?

- ▶ Uma linguagem de programação – Universidade de Potsdam (Alemanha)  
Universität Potsdam – 1999
- ▶ **Potassco**, the **P**otsdam **A**nswer **S**et **S**olving **C**ollection –  
<https://potassco.org/>
- ▶ <https://github.com/potassco-asp-course/> ... muitos slides
- ▶ Muitos slides, referências, códigos, vídeos no Youtube com os autores, etc
- ▶ Este material e os códigos apresentados:  
[https://github.com/claudiosa/CCS/tree/master/asp\\_Answer\\_Set\\_Programming](https://github.com/claudiosa/CCS/tree/master/asp_Answer_Set_Programming)
- ▶ Livros sobre ASP:

## Os livros:

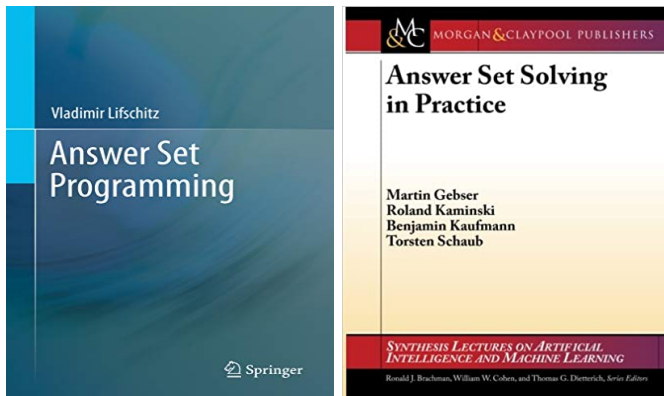


Figura: Estou usando o do Vladimir

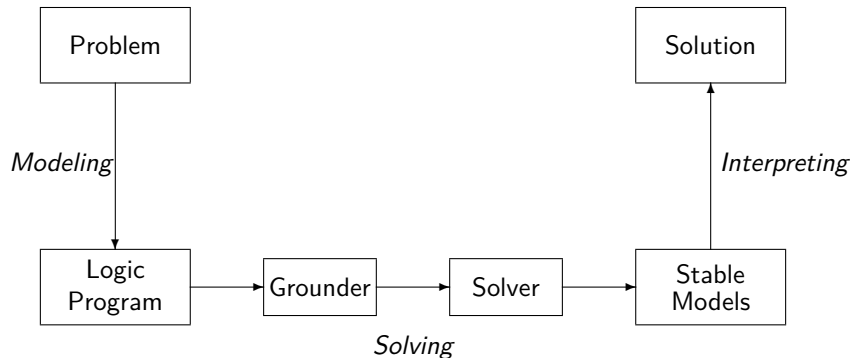
# Características

- ▶ Mais declarativa que Prolog e seus predecessores (**apenas, e quase nada mais, tem uma sintaxe que lembra Prolog**)
- ▶ Raízes em várias lógicas, incluindo as que tratam de informações incompletas: LP, LPO, *default*, circunscrição (suposição do mundo-fechado) e negação como falha, (auto-epistêmica)
- ▶ Usa o conceito de **modelo estável**: *semântica bem-fundamentada e ramificação*
- ▶ Uso: problemas combinatoriais baseados em conhecimento declarativo – faremos um exemplo
- ▶ Consistem de **decisões e restrições**
- ▶ Tudo isto na ordem de milhões!
- ▶ Na indústria: desde gerenciador de pacotes do Debian a sistemas da NASA

# Nesta apresentação

- ▶ A linguagem ASP com o sistema *clingo*
- ▶ *clingo* = *gringo* + *clasp*
- ▶ Há outras ramificações: *clingocon*, *aspcud* e *asprin*
- ▶ Alguns elementos da linguagem e um exemplo

# Modelagem, aterramento, e resolução



Fonte: <https://github.com/potassco-asp-course/>



# 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 <sup>1</sup>

*Fui há uma festa e apresentado há três casal. Os maridos tinham profissões e esposas distintas (até então o que se sabe desta estória). Após alguns "goles" me confundi quem era casado com quem, e as profissões. Apenas lembro de alguns fatos, então me ajude descobrir quem são estes casais, com base nos seguintes dados que me lembro:*

1. O médico é casado com a Maria;
2. O Paulo é advogado;
3. Patrícia não é casada com Paulo;
4. Carlos não é médico.

---

<sup>1</sup>Retirado da revista Coquetel: Problemas de Lógica 

# Problema dos maridos, esposas e profissões

Os demais dados são:

$H = \{ \text{carlos, luiz, paulo} \}$

$M = \{ \text{maria, lucia, patricia} \}$

$P = \{ \text{advogado, medico, engenheiro} \}$

# Problema dos maridos, esposas e profissões

Os demais dados são:

$H = \{ \text{carlos, luiz, paulo} \}$

$M = \{ \text{maria, lucia, patricia} \}$

$P = \{ \text{advogado, medico, engenheiro} \}$

- ▶ Enfim, me ajude a encontrar que são os pares e as profissões dos maridos?
- ▶ A curiosidade deste problema: embora tenhamos poucas informações, este **só tem uma resposta possível!**

# Abordagem ingênua (*naive*) do código do Prolog I

```
1 % Modelagem: ((H1,M1,P1), (H2,M2,P2), (H3,M3,P3))
2 % FATOS
3 prof( advogado ; medico ; engenheiro ).
4 esposas( patricia ; lucia ; maria ).
5 %* encerrado os fatos do problema %*
6 %% Exemplos de regras:
7 %% no minimo 3 e no maximo 7 elementos por conjunto
8 3 { casais(carlos,M1,P1) ;
9     casais(luis,M2,P2) ;
10     casais(paulo,M3,advogado) } 7
11
12 %*
13 2 ou mais elementos por respostas
14 2 {casais(carlos,M1,P1) ; casais(luis,M2,P2) ;
15     casais(paulo,M3,advogado)}
16 1 respostas por conjunto ... neste caso 1 por casal
17 1 {casais(carlos,M1,P1) ; casais(luis,M2,P2) ;
18     casais(paulo,M3,advogado)} 1
19 %*
20 :-
21     esposas(M1),
22     esposas(M2),
23     esposas(M3),
```

## Abordagem ingênua (*naive*) do código do Prolog II

```
24     prof(P1),  
25     prof(P2),  
26     M3 != patricia, %%Patricia NAO eh casada com Paulo  
27     P1 != medico,   %%Carlos NAO eh medico  
28     P2 = medico,  
29     M2 = maria, %%0 medico eh casado com a Maria  
30     %% lendo o texto, P2 eh medico, pois P3 eh advogado  
31     P1 != advogado,  
32     P2 != advogado,  
33     P1 != P2,  
34     M1 != M2,      %% varios diferentes  
35     M1 != M3,  
36     M2 != M3.  
37  
38 %% Imprimindo a saida:  
39 #show casais/3.
```

## Saída e reflexões

```
$ clingo ../07_lesson.lp 0
```

```
.....
```

```
Answer: 1
```

```
casais(paulo,lucia,advogado)
```

```
casais(luis,maria,medico) casais(carlos,patricia,engenheiro)
```

```
SATISFIABLE
```

```
Models          : 1
```

```
.....
```

Reflexões:



# Saída e reflexões

```
$ clingo ../07_lesson.lp 0
.....
Answer: 1
casais(paulo,lucia,advogado)
casais(luis,maria,medico) casais(carlos,patricia,engenheiro)
SATISFIABLE
```

```
Models          : 1
.....
```

Reflexões:

- ▶ Foi uma grande vitória inicial ... mas há uma caminhada....
- ▶ Nesta sua abordagem, voce não vai longe, há algo exponencial subjacente! – *Potassco team*

# Abordagem ASP I

*Aprender sempre: idéias dos códigos do Hakan*

```
1 % clingo 08_lesson.lp 0
2 %%% BASE DE FATOS ...
3 prof( advogado ; medico ; engenheiro ).
4 esposas( patricia ; lucia ; maria ).
5 homens( luis ; paulo ; carlos ).
6
7 %*
8   Encerrado os fatos do problema
9 %*
10 %%% EQUIVALENTE UM TODOS DIFERENTES ....
11 1 { casal(H, M, P) : homens(H), esposas(M) } 1 :- prof(P).
12 %%% UM CONJUNTO COM TODOS OS PARES repetidos ou nao de H e M,
13 %%% mas profissoes DIFERENTES
14 %%% AND
15 1 { casal(H, M, P) : homens(H), prof(P) } 1 :- esposas(M).
16 %%% idem para esposas diferentes
17 %%% AND
18 1 { casal(H, M, P) : esposas(M), prof(P) } 1 :- homens(H).
19 %%% idem para homens diferentes
20 %* UMA COMBINACAO destes 3 conjuntos fornece um
21   ALL-DIFFERENT, classico
22   Estas 3 regras foram a chave deste problema
```

# Abordagem ASP II

```
23 *%
24 %% AS RESTRICOES
25
26 %% NOT ALLOWED
27 :- casal(carlos,_,medico).           %% Carlos NAO eh medico
28 :- casal(paulo, patricia, _).       %% Patricia NAO eh casada com Pa
29
30 %% ALLOWED
31 :- not casal(paulo , _ , advogado). %% O Paulo EH advogado
32 :- not casal(_ , maria , medico).   %% O medico EH casado com a Mari
33
34 #show casal/3.
35
36 %*
37 THE END -- FIM
38 *%
```

Um código declarativo e limpo!

# Abordagem ASP - Saída

```
$ clingo 08_lesson.lp 0
.....
Answer: 1
casal(luis,maria,medico) casal(paulo,lucia,advogado)
    casal(carlos,patricia,engenheiro)
SATISFIABLE

Models          : 1
.....
```

# Conclusões Parciais

- ▶ Sozinho, sem uma fundametação lógico-matemática, hum ...  
diria que há uma barreira inicial

# Conclusões Parciais

- ▶ Sozinho, sem uma fundametação lógico-matemática, hum ... diria que há uma barreira inicial
- ▶ Como desafio intelectual de uma linguagem, completamente diferente, fantástica!

# Conclusões Parciais

- ▶ Sozinho, sem uma fundametação lógico-matemática, hum ... diria que há uma barreira inicial
- ▶ Como desafio intelectual de uma linguagem, completamente diferente, fantástica!
- ▶ Da dificuldade  $\Rightarrow$  há uma oportunidade!

# Epílogo – Negações dos Predicados

```
1 p(3..7).  
2 q(5..9).  
3 s(X) :- X = 1..10, p(X), not q(X).  
4  
5 #show s/1.  
6 %#show p/1.  
7 %#show q/1.
```

Saída:



# Epílogo – Negações dos Predicados

```
1 p(3..7).  
2 q(5..9).  
3 s(X) :- X = 1..10, p(X), not q(X).  
4  
5 #show s/1.  
6 %#show p/1.  
7 %#show q/1.
```

Saída:

```
$ clingo 12_lesson.lp 0  
.....  
Answer: 1  
s(3) s(4)  
SATISFIABLE
```

Embora **p** tenha modelos estáveis de 3 a 7, e **q** de 5 a 9.

# Próximos Passos

- ▶ Estudar o *clingocon*, estende o *clingo* com *Constraint Programming*
- ▶ Mais códigos e fazer os simples
- ▶ Ver outros tipos de problemas que ASP seja aderente: planejamento, etc.
- ▶ Agradecimentos: Vladimir, Hakan, e Roland Kaminski

## Contato e Comentários:

- ▶ <https://claudiocesars.wordpress.com/>
- ▶ <https://github.com/claudiosa>
- ▶ Neste git, repositório CCS  $\Rightarrow$  asp...
- ▶ Email: [ccs1664@gmail.com](mailto:ccs1664@gmail.com)
- ▶ *Thank you so much!*