

Programação em Lógica

Prof. A. G. Silva

26 de outubro de 2017

Resolução de problemas combinatórios

Resolvendo o quebra-cabeça SUDOKU

Lógica de programação com restrições

Jogando Sudoku

		5	4		6	1		
	8				1		9	
		4		1		5		
	7			9			2	
		6		8		3		
	2						7	
			5		3	6		

Considerando um problema mais simples

		4	
	2		
		1	
	3		

Modelagem em Prolog usando permutações de lista

Cada linha deve ser uma permutação de [1, 2, 3, 4]

Cada coluna deve ser uma permutação de [1, 2, 3, 4]

Cada quadrado 2x2 deve ser uma permutação de [1, 2, 3, 4]

Representação do quadro como lista de listas

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

```
[ [A,B,C,D],  
  [E,F,G,H],  
  [I,J,K,L],  
  [M,N,O,P] ]
```

Predicado sudoku construído como restrições de permutações simultâneas

```
teste :-
    sudoku( [[_, _, 4, _],
             [_, 2, _, _],
             [_, _, 1, _],
             [_, 3, _, _]] ).

sudoku( [[X11,X12,X13,X14],
         [X21,X22,X23,X24],
         [X31,X32,X33,X34],
         [X41,X42,X43,X44]] ) :-
    %linhas
    permutation([X11,X12,X13,X14],[1,2,3,4]),
    permutation([X21,X22,X23,X24],[1,2,3,4]),
    permutation([X31,X32,X33,X34],[1,2,3,4]),
    permutation([X41,X42,X43,X44],[1,2,3,4]),
    %colunas
    permutation([X11,X21,X31,X41],[1,2,3,4]),
    permutation([X12,X22,X32,X42],[1,2,3,4]),
    permutation([X13,X23,X33,X43],[1,2,3,4]),
    permutation([X14,X24,X34,X44],[1,2,3,4]),
    %quadrados
    permutation([X11,X12,X21,X22],[1,2,3,4]),
    permutation([X13,X14,X23,X24],[1,2,3,4]),
    permutation([X31,X32,X41,X42],[1,2,3,4]),
    permutation([X33,X34,X43,X44],[1,2,3,4]),
    %exibicao
    write(X11), write(' '), write(X12), write(' '),
    write(X13), write(' '), write(X14), nl,
    write(X21), write(' '), write(X22), write(' '),
    write(X23), write(' '), write(X24), nl,
    write(X31), write(' '), write(X32), write(' '),
    write(X33), write(' '), write(X34), nl,
    write(X41), write(' '), write(X42), write(' '),
    write(X43), write(' '), write(X44), nl.
```

Aumento da escala de forma óbvia para 3×3

X11	X12	X13	X14	X15	X16	X17	X18	X19
X21	X22	X23	X24	X25	X26	X27	X28	X29
X31	X32	X33	X34	X35	X36	X37	X38	X39
X41	X42	X43	X44	X45	X46	X47	X48	X49
X51	X52	X53	X54	X55	X56	X57	X58	X59
X61	X62	X63	X64	X65	X66	X67	X68	X69
X71	X72	X73	X74	X75	X76	X77	X78	X79
X81	X82	X83	X84	X85	X86	X87	X88	X89
X91	X92	X93	X94	X95	X96	X97	X98	X99

Força bruta é impraticavelmente lenta

Há uma quantidade muito grande de disposições na grade:
 $6670903752021072936960 \approx 6.671 \times 10^{21}$

Nossa abordagem atual não codifica as inter-relações
entre as restrições

Para obter mais informações sobre enumeração Sudoku:
<http://www.afjarvis.staff.shef.ac.uk/sudoku/>

Programas em Prolog podem ser vistos como problemas de satisfação de restrições

Prolog é limitado à restrição de igualdade única:
- Dois termos devem unificar

Podemos generalizar para incluir outros tipos de limitação

Isso leva à Programação de Lógica de Restrições
(Constraint Logic Programming)
- Um modo de resolver problemas como o Sudoku

Consideram-se valores de variáveis em um domínio e com restrições

Dado:

- um conjunto de variáveis
- o domínio de cada variável
- restrições sobre estas variáveis

Pretende-se encontrar:

- Uma atribuição de valores a variáveis que satisfaçam às restrições

Sudoku pode ser expressado como restrições

Primeiramente, nós expressamos as variáveis e seus domínios:

$A \in \{1,2,3,4\}$

$C \in \{1,2,3,4\}$

$E \in \{1,2,3,4\}$

$G \in \{1,2,3,4\}$

$I \in \{1,2,3,4\}$

$K \in \{1,2,3,4\}$

$M \in \{1,2,3,4\}$

$O \in \{1,2,3,4\}$

$B \in \{1,2,3,4\}$

$D \in \{1,2,3,4\}$

$F \in \{1,2,3,4\}$

$H \in \{1,2,3,4\}$

$J \in \{1,2,3,4\}$

$L \in \{1,2,3,4\}$

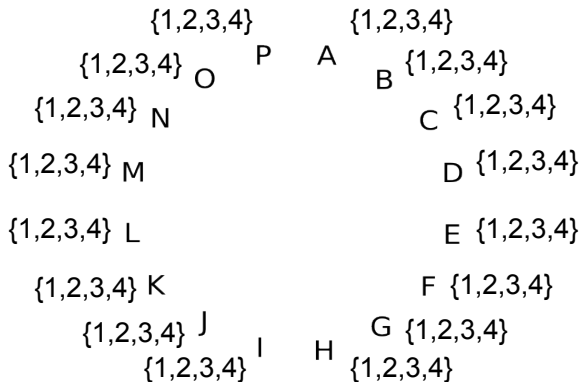
$N \in \{1,2,3,4\}$

$P \in \{1,2,3,4\}$

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

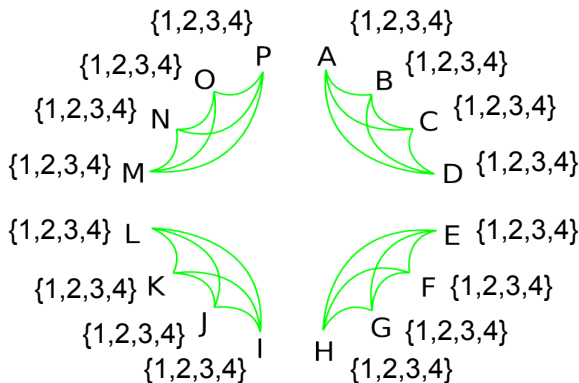
Expressando o Sudoku como um grafo de restrições

As arestas representam desigualdades entre variáveis



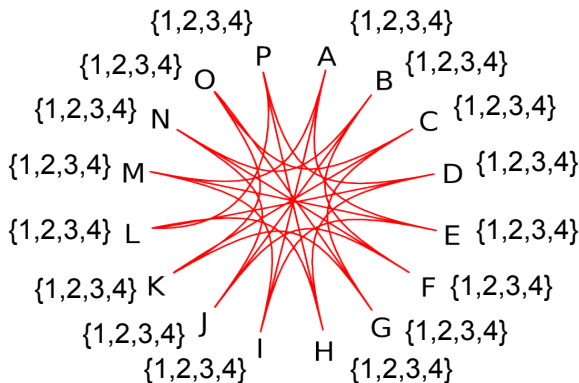
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Restrição: todas as variáveis em cada linha são diferentes



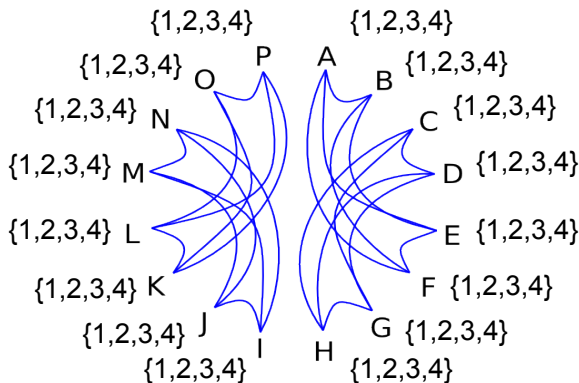
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Restrição: todas as variáveis em cada coluna são diferentes



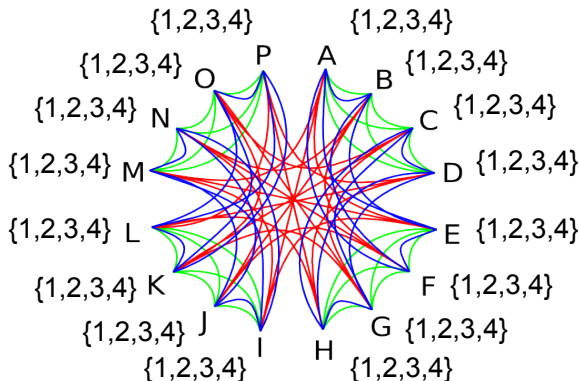
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Restrição: todas em cada quadrado são diferentes



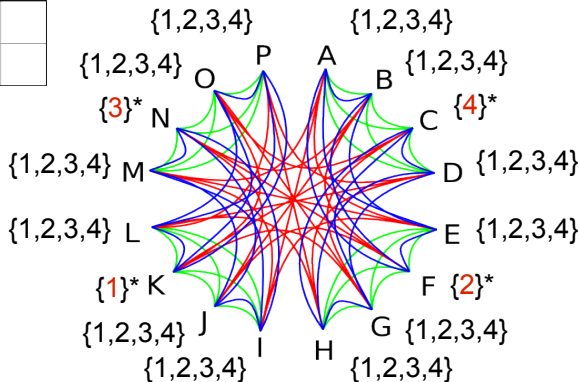
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Exibição de todas as restrições juntas



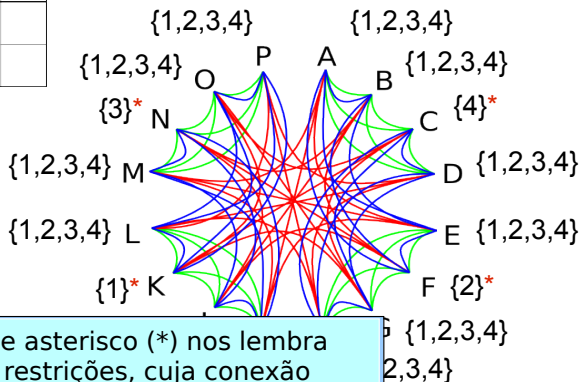
Redução de domínio de acordo com os valores iniciais

		4	
	2		
		1	
	3		



Ao alterar o domínio, atualizam-se suas restrições

		4	
	2		
		1	
	3		

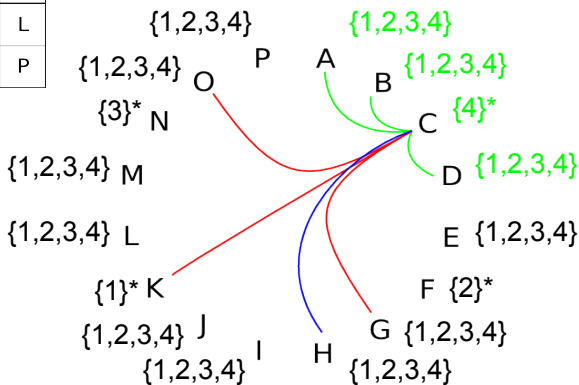


A notação de asterisco (*) nos lembra de todas as restrições, cuja conexão para esta variável precisa de atualização

Atualização de restrições conectadas a C

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

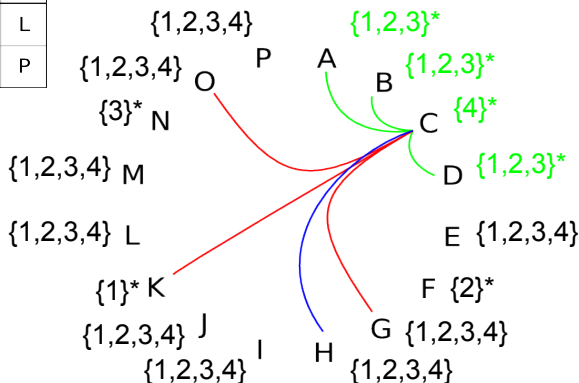
Remove-se o 4 a partir do domínio de A, B e D



Atualização de restrições conectadas a C

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

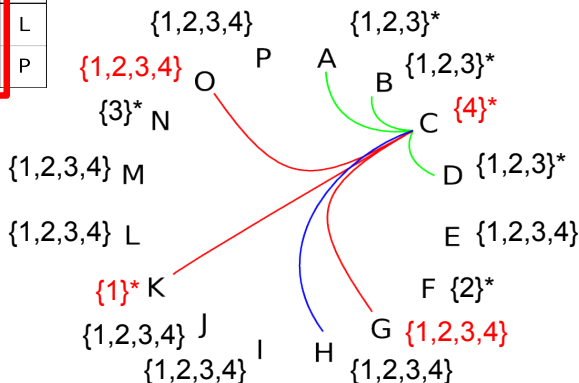
Adiciona-se asteriscos a A, B e D
... mas trataremos deles posteriormente



Atualização de restrições conectadas a C

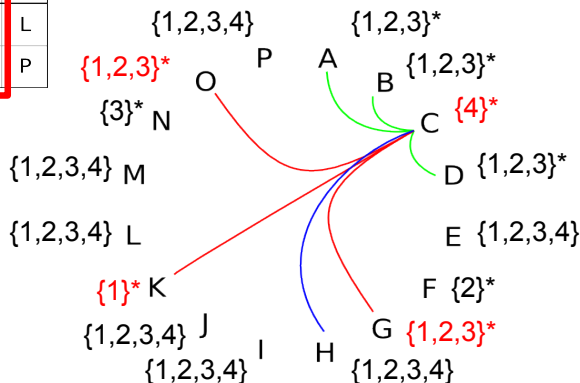
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Examina-se as restrições das colunas



Atualização de restrições conectadas a C

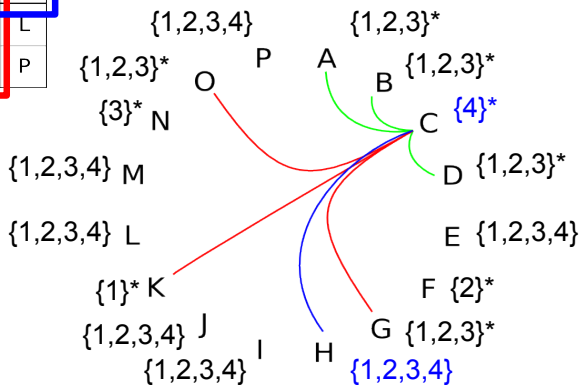
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Atualização de restrições conectadas a C

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

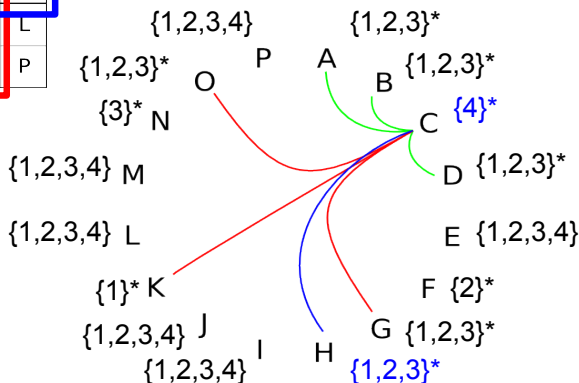
Note que D e G já têm seus domínios atualizados



Atualização de restrições conectadas a C

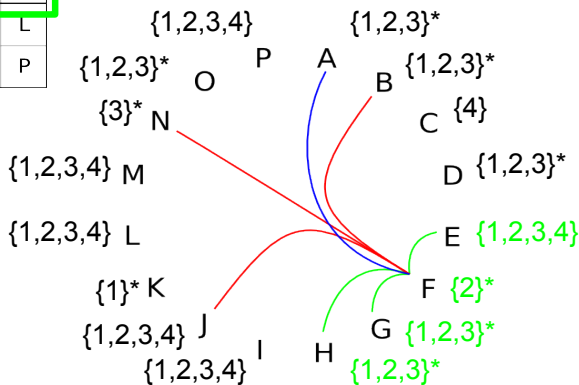
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Esgotamos as restrições de C por hora



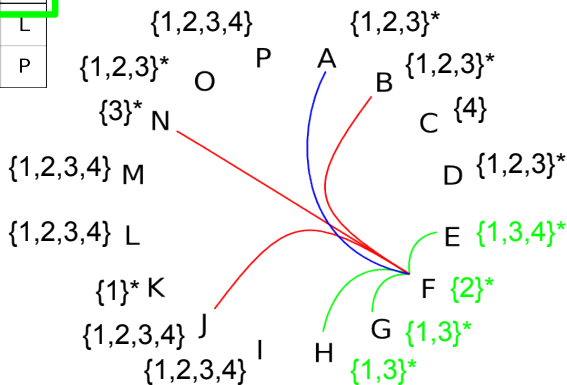
Atualização de restrições conectadas a F

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



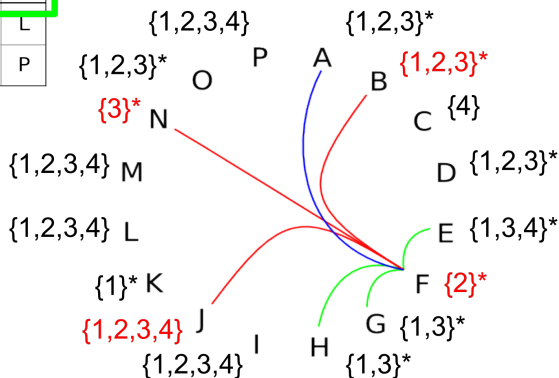
Atualização de restrições conectadas a F

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



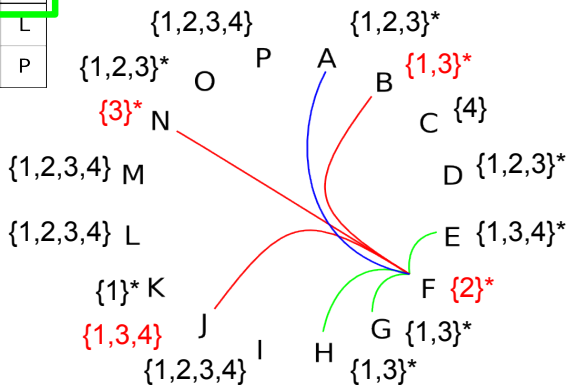
Atualização de restrições conectadas a F

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



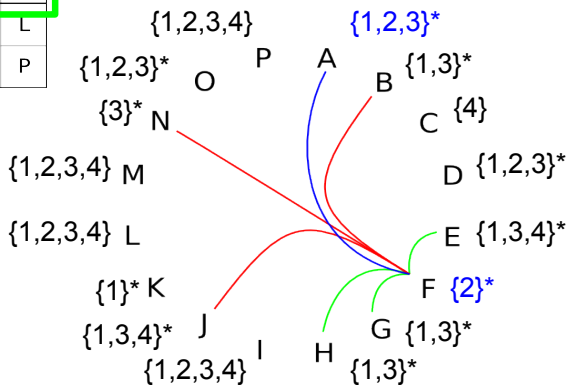
Atualização de restrições conectadas a F

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Atualização de restrições conectadas a F

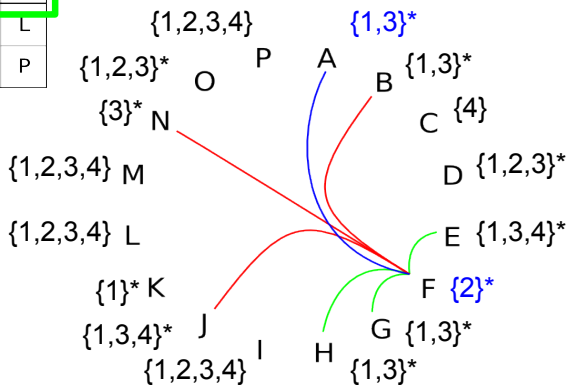
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Atualização de restrições conectadas a F

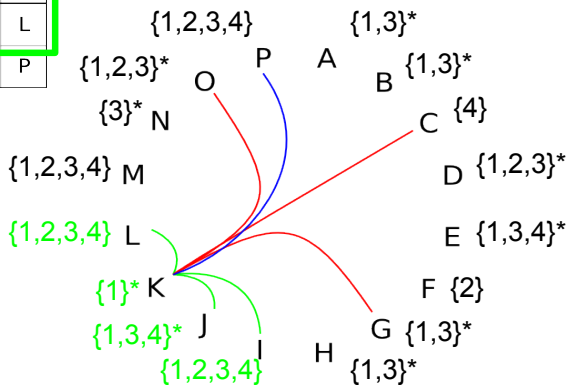
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Esgotamos as restrições de F por hora



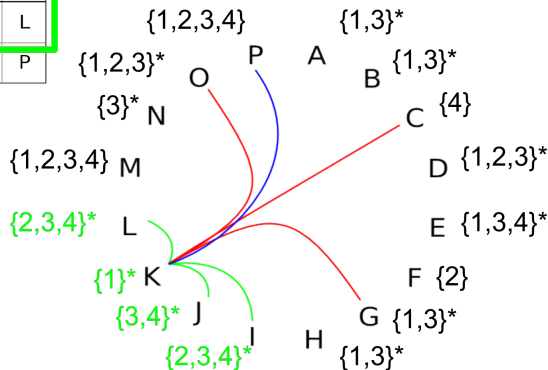
Atualização de restrições conectadas a K

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



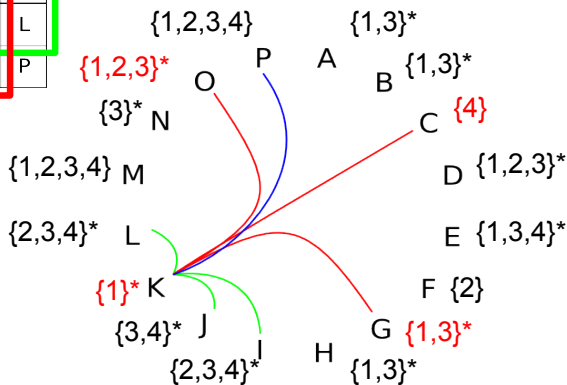
Atualização de restrições conectadas a K

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



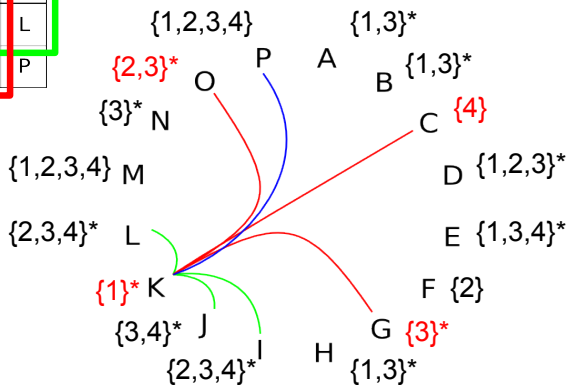
Atualização de restrições conectadas a K

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



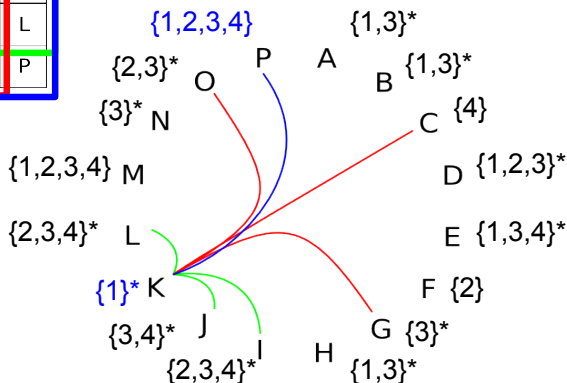
Atualização de restrições conectadas a K

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



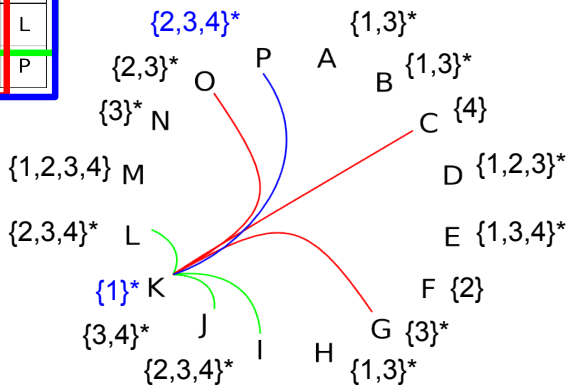
Atualização de restrições conectadas a K

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Atualização de restrições conectadas a K

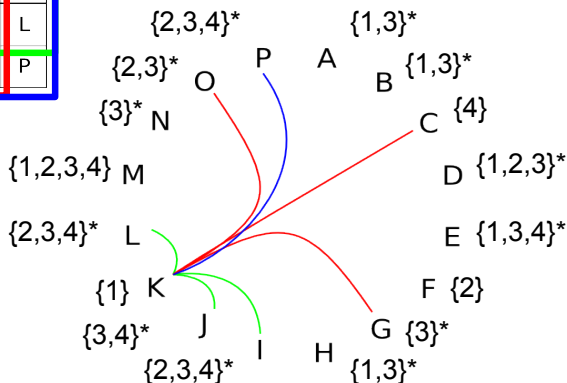
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Atualização de restrições conectadas a K

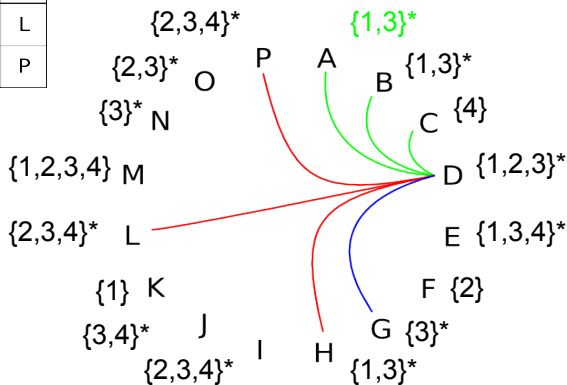
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Esgotamos as restrições de K por hora



Atualização de restrições conectadas a D

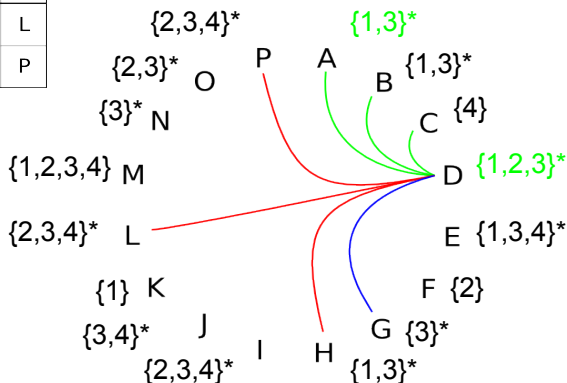
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Atualização de restrições conectadas a D

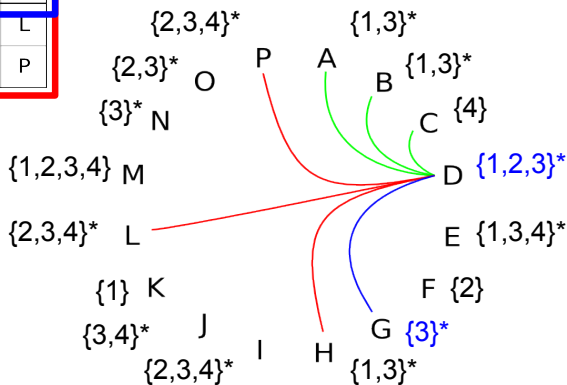
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Nenhum valor pode ser eliminado diretamente (mas devemos observar a resposta)



Atualização de restrições conectadas a D

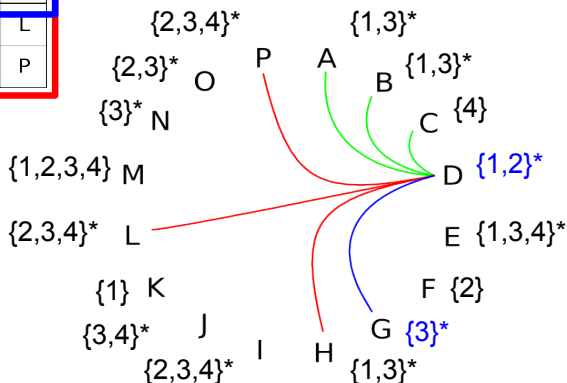
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P



Alteração pode ocorrer em domínio de origem

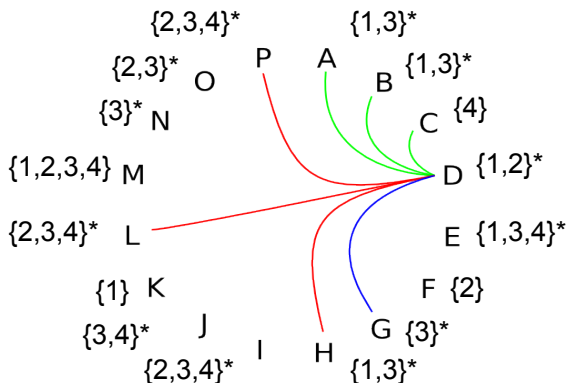
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

O único 3 em G elimina o 3 de D



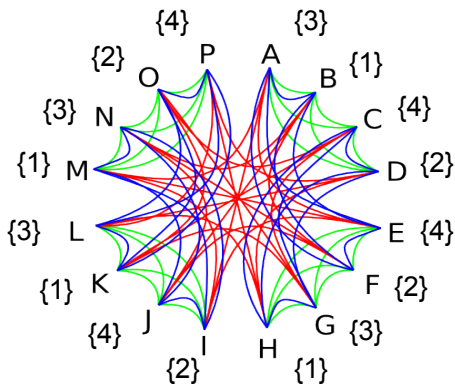
Alteração pode ocorrer em domínio de origem

Se o domínio de origem modificar, marca-se todas as suas restrições para uma nova atualização



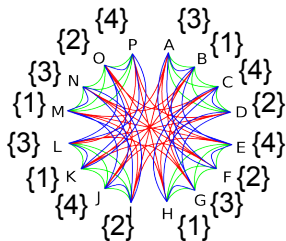
Convergência do algoritmo (nenhuma alteração ocorrerá)

Por que o algoritmo converge eventualmente?



Resultado 1: domínios de valor único

Encontra-se uma única solução para o problema



3	1	4	2
4	2	3	1
2	4	1	3
1	3	2	4

Resultado 2: Alguns domínios vazios

As restrições mostram-se inconsistentes
- Não há nenhuma solução para este problema

Variáveis

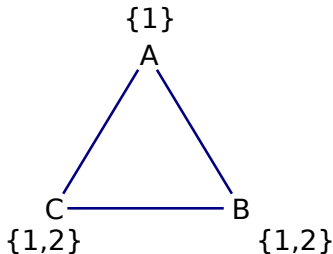
$$A \in \{1\}$$

$$B \in \{1,2\}$$

$$C \in \{1,2\}$$

Restrições

$$A \neq B, A \neq C, B \neq C$$



Resultado 2: Alguns domínios vazios

As restrições mostram-se inconsistentes
- Não há nenhuma solução para este problema

Variáveis

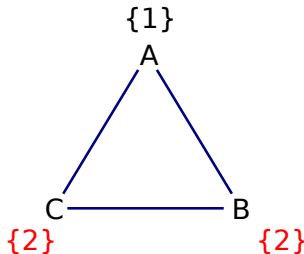
$A \in \{1\}$

$B \in \{1,2\}$

$C \in \{1,2\}$

Restrições

$A \neq B, A \neq C, B \neq C$



Resultado 2: Alguns domínios vazios

As restrições mostram-se inconsistentes
- Não há nenhuma solução para este problema

Variáveis

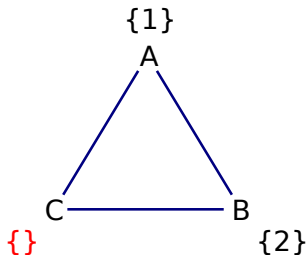
$A \in \{1\}$

$B \in \{1,2\}$

$C \in \{1,2\}$

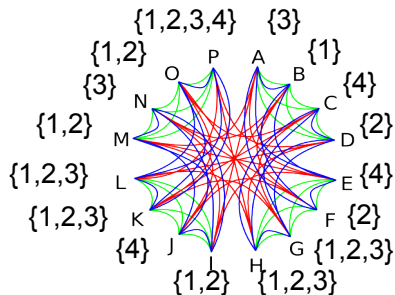
Restrições

$A \neq B, A \neq C, B \neq C$



Resultado 3: Alguns domínios multivalorados

		4	
	2		
	3		



Apesar de que nem todas as combinações destas possibilidades de atribuição de variável são soluções globais...

Resultado 4: Rotulação (etiquetagem) de hipóteses

Para encontrar soluções globais a partir de domínios pequenos, efetua-se a hipótese de solução em um domínio e propagam-se as mudanças

Backtrack se algo der errado

Usando CLP (*Constraint Logic Programming*) em Prolog

```
:- use_module(library(clpfd)).
```

```
teste :-
```

```
    sudoku2( [[_, _, 4, _],
               [_, 2, _, _],
               [_, _, 1, _],
               [_, 3, _, _]] ).
```

```
valid4(L) :- L ins 1..4, all_distinct(L).
```

```
sudoku2( [[X11,X12,X13,X14],
           [X21,X22,X23,X24],
           [X31,X32,X33,X34],
           [X41,X42,X43,X44]] ) :-
```

```
%linhas
```

```
valid4([X11,X12,X13,X14]),
valid4([X21,X22,X23,X24]),
valid4([X31,X32,X33,X34]),
valid4([X41,X42,X43,X44]),
```

```
%colunas
```

```
valid4([X11,X21,X31,X41]),
valid4([X12,X22,X32,X42]),
valid4([X13,X23,X33,X43]),
valid4([X14,X24,X34,X44]),
```

```
%quadrados
```

```
valid4([X11,X12,X21,X22]),
valid4([X13,X14,X23,X24]),
valid4([X31,X32,X41,X42]),
valid4([X33,X34,X43,X44]),
```

```
%rotulacao
```

```
labeling([], [X11,X12,X13,X14,X21,X22,X23,X24,
               X31,X32,X33,X34,X41,X42,X43,X44]),
```

```
%exibicao
```

```
write(X11), write(' '), write(X12), write(' '),
write(X13), write(' '), write(X14), nl,
write(X21), write(' '), write(X22), write(' '),
write(X23), write(' '), write(X24), nl,
write(X31), write(' '), write(X32), write(' '),
write(X33), write(' '), write(X34), nl,
write(X41), write(' '), write(X42), write(' '),
write(X43), write(' '), write(X44), nl.
```

Referências

- Baseado no material da Universidade de Cambridge (com adaptações no programa para SWI-Prolog):
<https://www.cl.cam.ac.uk/teaching/0809/Prolog/>
- Outros problemas interessantes:
<https://github.com/triska/clpfd>