

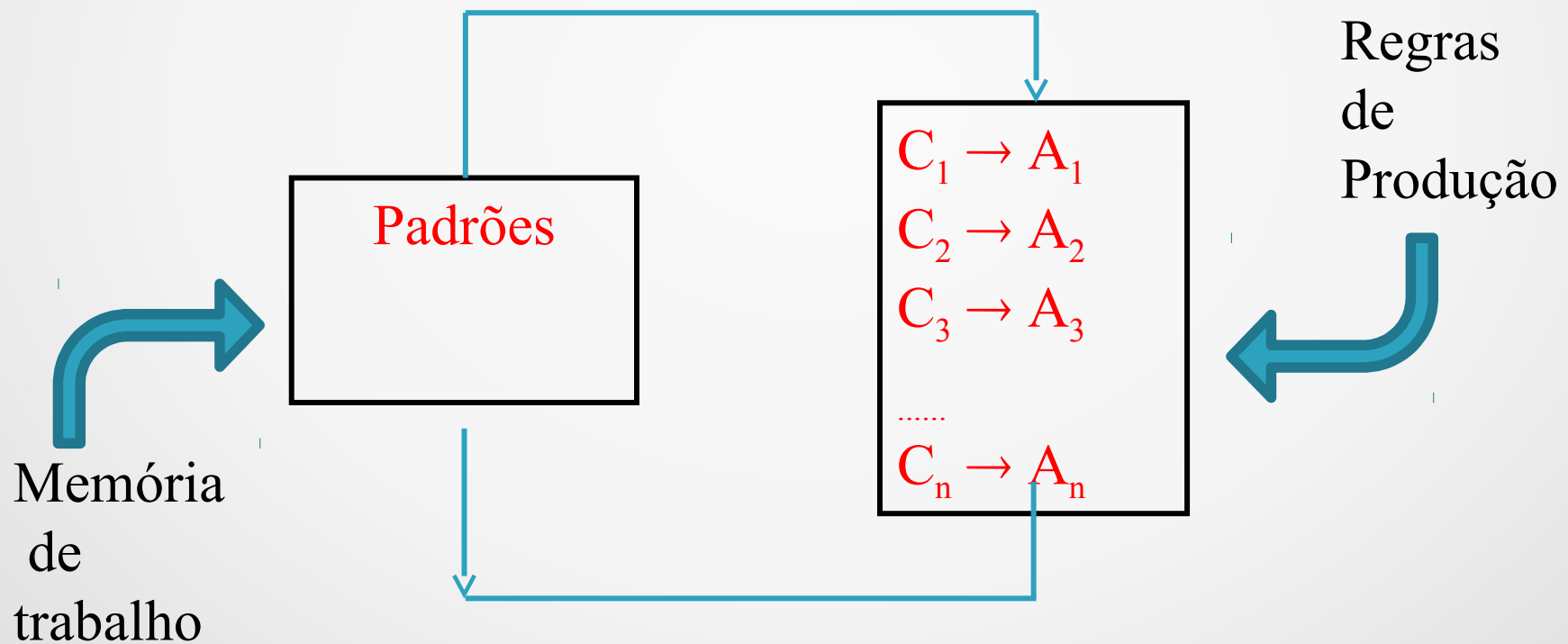
Aula 3 – Sistemas de Produção

Parte 2 – Métodos de Inferência
22705/1001336 - Inteligência Artificial
2019/1 - Turma A
Prof. Dr. Murilo Naldi

naldi@dc.ufscar.br

Sistemas de produção

- Vimos que um sistema de produção possui um ciclo reconhecimento-ação gera uma sequencia de aplicação de regras.



Controle

- O ciclo reconhecimento-ação gera uma sequência de aplicação de regras.
- Estratégias de Raciocínio
 - Busca Dirigida por **Dados** (*Data-driven*)
 - Inicia com a descrição do problema (dados: axiomas lógicos, sintomas de uma doença, dados a serem interpretados), aplica movimentos até encontrar a solução
 - Busca Dirigida por **Objetivos** (*Goal-driven*)
 - Inicia com um objetivo (teorema, diagnóstico, classe de um objeto) e tenta provar sua validade

Estratégias

- **Encadeamento Progressivo** (*Forward chaining*)
 - Implementa a busca dirigida por dados
 - Compatível com o ciclo reconhecimento-ação apresentado.
- **Encadeamento Regressivo** (*Backward chaining*)
 - Implementa a busca dirigida por objetivos
 - O ciclo-reconhecimento-ação deve ser modificado.

Encadeamento Progressivo

- Implementa a busca dirigida por dados (*data driven*)
- Podemos entender o encadeamento progressivo como um processo de inferência do cálculo proposicional
 - Aplica *modus ponens* em regras da base de conhecimento
 - Gera novas sentenças, em direção a prova.
- Usando unificação do conteúdo da memória de trabalho com o antecedente das regras quando os símbolos são iguais.

West é criminoso ou não?

- “A lei diz que é crime um americano vender armas a nações hostis. O país Nono, inimigo da América, tem alguns mísseis vendidos pelo Coronel West, um americano.”
- Como modelar a base de conhecimento para o caso do Coronel West?

Modelagem

- “A lei diz que é crime um americano vender armas a nações hostis”
 - **(1)** $americano(X) \wedge arma(Y) \wedge hostil(Z) \wedge vende(X,Z,Y) \rightarrow criminoso(X)$
- “Nono ... tem alguns mísseis”
 - $\exists X possui(nono,X) \wedge missil(X)$
 - **(2)** $possui(nono, m_1)$
 - **(3)** $missil(m_1)$
- Quando uma sentença possui apenas conjunções e não há variável em comum é possível (e aconselhável) dividi-la em sentenças atômicas

Modelagem

- “...(os mísseis) foram vendidos pelo Coronel West”
- **(4)** $\text{possui}(\text{nono}, X) \wedge \text{míssil}(X) \rightarrow \text{vende}(\text{west}, \text{nono}, X)$
- Sabemos que mísseis são armas
- **(5)** $\text{míssil}(X) \rightarrow \text{arma}(X)$
- Um inimigo é hostil
- **(6)** $\text{inimigo}(X, \text{usa}) \rightarrow \text{hostil}(X)$

Modelagem

- “...(West) é americano.”
 - (7) *americano(west)*
- “O país Nono, inimigo da américa...”
 - (8) *inimigo(nono,usa)*
- As sentenças com atômicas são a 2, 3, 7 e 8
- As sentenças com implicação ou regras são a 1, 4, 5 e 6

Como aplicar

1. Utilize *Modus Ponens* em todas as sentenças **atômicas** para cada sentença com implicação (**regras**).
2. Se alguma estiver habilitada, uma nova sentença **atômica** é gerada e adicionada a base de conhecimento. Se for o objetivo para com sucesso, senão volta ao primeiro passo.
3. Se não houverem **regras** habilitadas, o algoritmo para e nada pode ser concluído.

Encadeamento Progressivo

Faça assim
na prova!

- Na primeira iteração:
- A regra **1** não é satisfeita
- A regra **4** é satisfeita quando unificada com as sentenças **2** e **3** com $\theta = \{X/m_1\}$. Como resultado, a sentença *vende(west,nono, m_1)* é adicionada a base de conhecimento como sentença de número **9**.

Encadeamento Progressivo

- Na primeira iteração:
- A regra **5** é satisfeita quando unificada com a sentença **3** com $\theta = \{X/m_1\}$. A sentença **(10)** *arma(m₁)* é adicionada.
- A regra **6** é satisfeita quando unificada com a sentença **8** com $\theta = \{X/nono\}$. A sentença **(11)** *hostil(nono)* é adicionada.

Encadeamento Progressivo

- Na segunda iteração:
- A regra **1** é satisfeita quando unificada com as sentenças 7, 10, 11 e 9 com $\theta = \{X/west, Y/m_1, Z/nono\}$. A sentença (12) *criminoso(west)* é o objetivo.
- **Desvantagem:** gera fatos irrelevantes, pois faz todas as inferências possíveis baseadas nos fatos conhecidos.

Representação em SP

- A execução de encadeamento progressivo em um SP pode ser representado por:
 - **Tabela de execução** – mostra a memória de trabalho, o conjunto de conflito e a regra disparada
 - **Grafo de execução** – mostra a relação entre antecedentes e consequentes das regras disparadas.
 - Subgrafo do grafo completo.
- São precisos os dados iniciais, o objetivo, e a estratégia de raciocínio e resolução de conflito.

Exemplo Encadeamento Progressivo

- ▶ Exemplo:

1. $p, q \Rightarrow o$

2. $r, s \Rightarrow p$

3. $w, r \Rightarrow p$

4. $t, u \Rightarrow q$

5. $v \Rightarrow s$

6. $i \Rightarrow v$

7. $i \Rightarrow r$

8. $i \Rightarrow q$

- Resolução de Conflito: regra menos usada até aquele ponto

Exemplo Encadeamento Progressivo

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	i	6,7,8	6

1. $p, q \Rightarrow o$

2. $r, s \Rightarrow p$

3. $w, r \Rightarrow p$

4. $t, u \Rightarrow q$

5. $v \Rightarrow s$

6. $i \Rightarrow v$

7. $i \Rightarrow r$

8. $i \Rightarrow q$

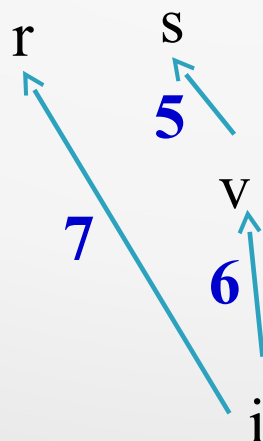
Dado inicial: i Objetivo: o
Resolução de conflito: regra menos usada até o momento



Exemplo Encadeamento Progressivo

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
1	i,v	5,6,7,8	5
2	i,v,s	5,6,7,8	7

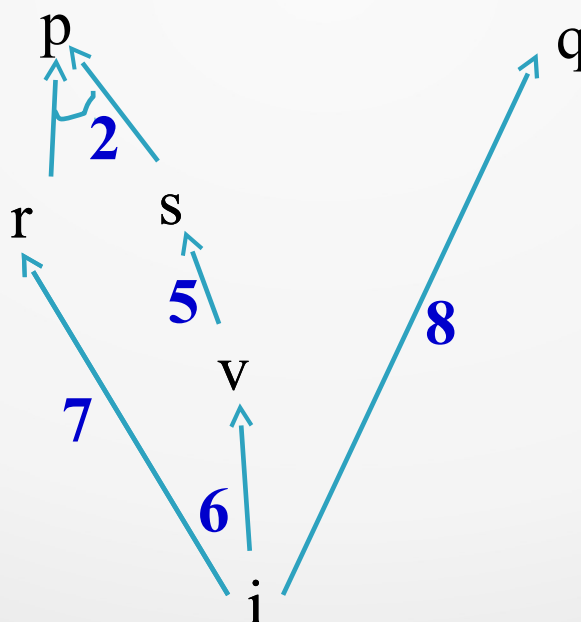
1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$



Exemplo Encadeamento Progressivo

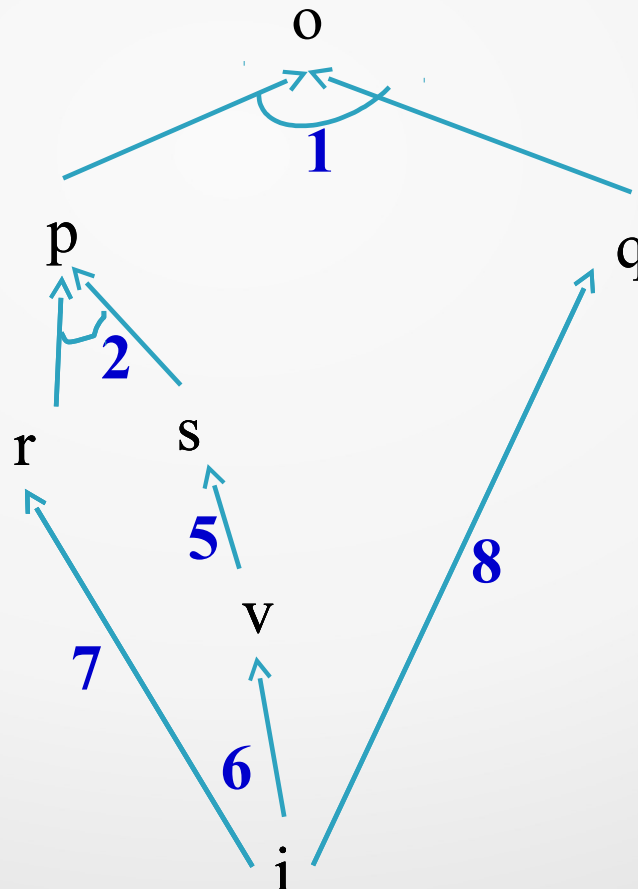
Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
3	i,v,s,r	2,5,6,7,8	2
4	i,v,s,r,p	2,5,6,7,8	8

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$



Exemplo Encadeamento Progressivo

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
5	i,v,s,r,p,q	1,2,5,6,7,8	1
6	i,v,s,r,p,q,o		Pare



1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$

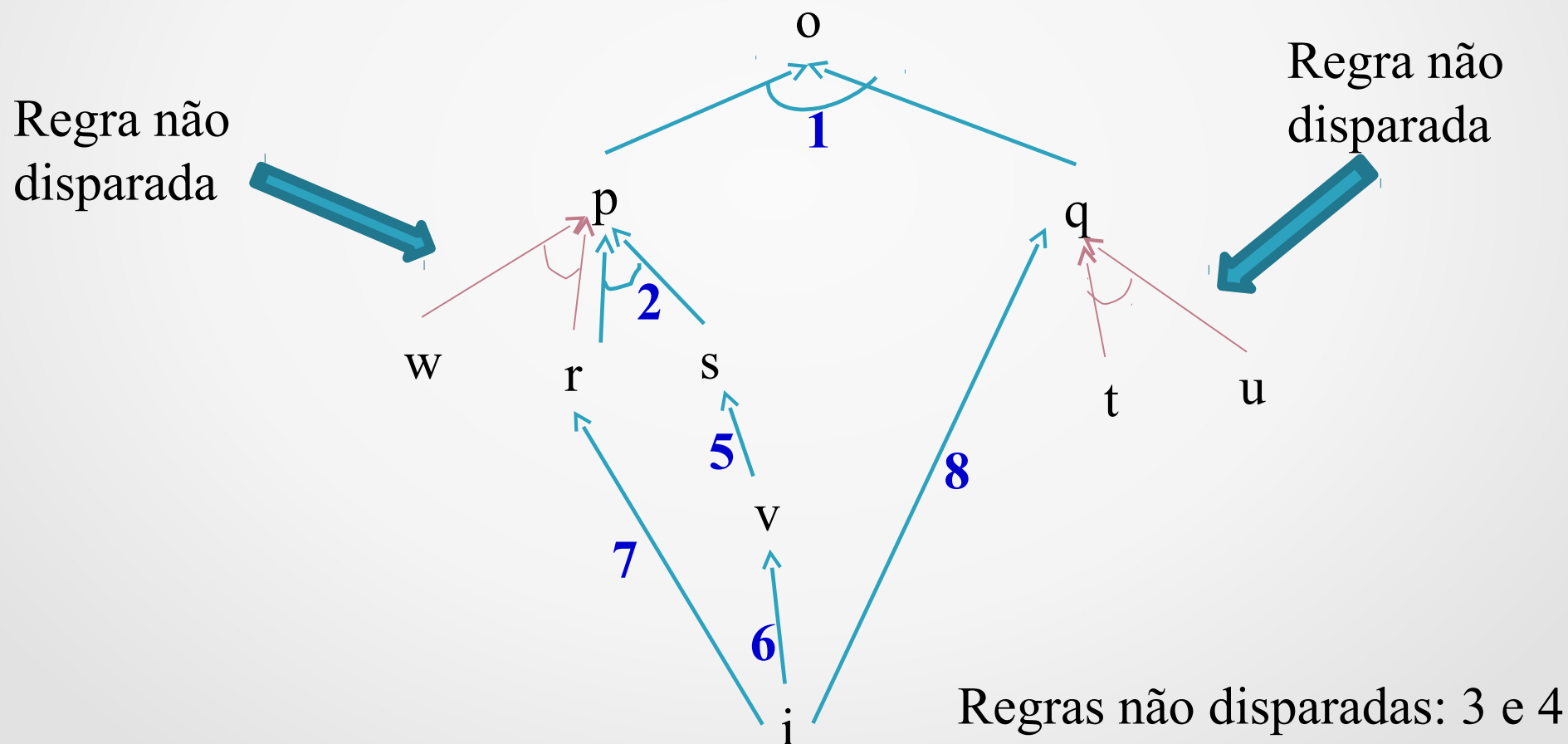
Exemplo Encadeamento Progressivo

Tabela de execução

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	i	6,7,8	6
1	i,v	5,6,7,8	5
2	i,v,s	5,6,7,8	7
3	i,v,s,r	2,5,6,7,8	2
4	i,v,s,r,p	2,5,6,7,8	8
5	i,v,s,r,p,q	1,2,5,6,7,8	1
6	i,v,s,r,p,q,o		Pare

Exemplo Encadeamento Progressivo

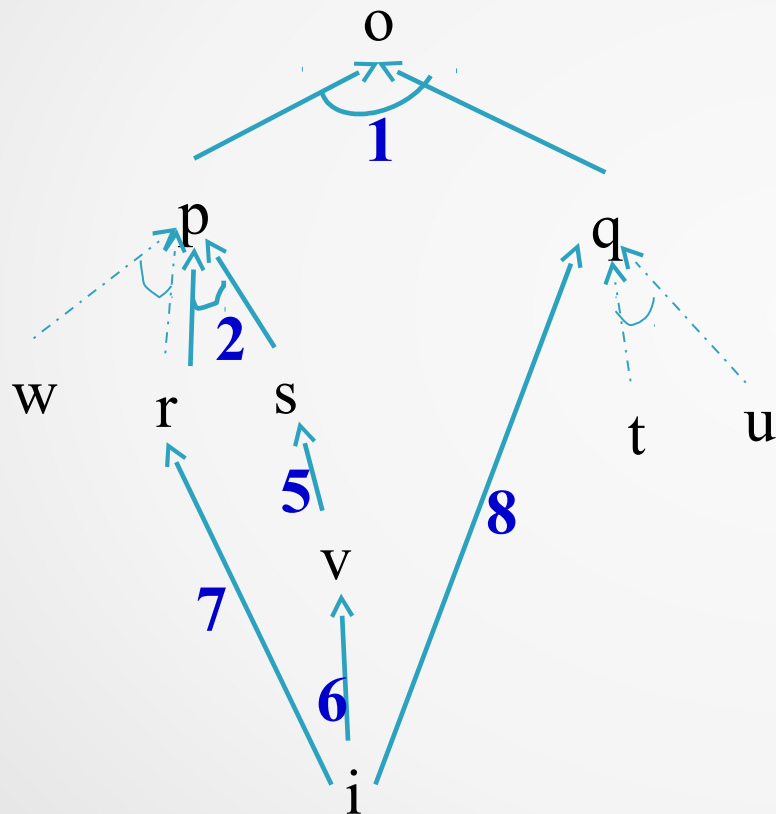
- ▶ Grafo de execução
- ▶ Sequência de regras disparadas: 6, 5, 7, 2, 8, 1



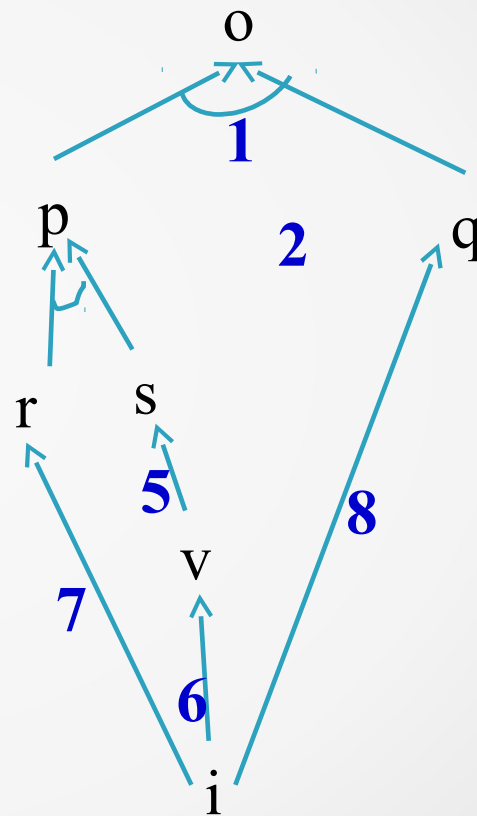
SP com encadeamento progressivo

- Grafo de solução – subgrafo do grafo de execução
 - É um caminho completo entre os dados iniciais e os objetivos
- Um grafo de execução pode conter mais de um grafo de solução
 - Neste exemplo, o grafo de solução é igual ao grafo de execução. Todas as regras disparadas fazem parte da mesma solução.

Grafo de execução e solução



Grafo de execução



Grafo de solução

Exemplo: i e w na memória de trabalho

Tabela de execução

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	i,w	6,7,8	6
1	i,w,v	5,6,7,8	5
2	i,w,v,s	5,6,7,8	7
3	i,w,v,s,r	2,3,5,6,7,8	2
4	i,w,v,s,r,p	2,3,5,6,7,8	3
5	i,w,v,s,r,p	2,3,5,6,7,8	8
6	i,w,v,s,r,p,q	1,2,3,5,6,7,8	1
6	i,w,v,s,r,p,q,o		Pare

Encadeamento para trás

- Implementa a busca dirigida por objetivos
- Inicia com um objetivo e caminha para trás, até encontrar fatos (dados iniciais) que validem o objetivo.
- Para usar a estratégia de encadeamento para trás, o ciclo de Reconhecimento-Ação deve ser adaptado.
- Utilizado em programação lógica (prolog).

Como aplicar

- O algoritmo inicia com uma lista de objetivos.
- Procura por uma cláusula com **consequente** que consiga ser unificada com o objetivo. Se não houver, o algoritmo para e nada pode ser inferido.
- A **premissa**, ou **antecedente**, dessa cláusula é adicionada ao início da lista de objetivos.
- Se a lista de objetivos ficar vazia, a prova é verdadeira. Senão volta ao passo 2.

Caso Coronel West

- Qual é o objetivo?
 - *criminoso(west)*
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(1)** *americano(X) ^ arma(Y) ^ hostil(Z) ^ vende(X,Z,Y) → criminoso(X)*
 - Unifica com o objetivo e $\theta = \{X/west\}$
 - Adiciona a premissa ao objetivo

Faça assim
na prova!

Caso Coronel West

- Qual o novo objetivo?
 - $\text{americano}(\text{west}) \wedge \text{arma}(Y) \wedge \text{hostil}(Z) \wedge \text{vende}(\text{west}, Z, Y)$
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - (7) $\text{americano}(\text{west})$
 - Unifica com o objetivo
 - Adiciona premissa ao objetivo (?)

Caso Coronel West

- Qual o novo objetivo?
 - $arma(Y) \wedge hostile(Z) \wedge vende(west, Z, Y)$
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(5)** $míssil(X) \rightarrow arma(X)$
 - Unifica com o objetivo e $\theta = \{X/Y\}$
 - Adiciona premissa ao objetivo

Caso Coronel West

- Qual o novo objetivo?
 - $hostil(Z) \wedge vende(west, Z, m_1)$
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(6)** $inimigo(X, usa) \rightarrow hostil(X)$
 - Unifica com o objetivo e $\theta = \{X/Z\}$
 - Adiciona premissa ao objetivo

Caso Coronel West

- Qual o novo objetivo?
 - $\text{inimigo}(Z, \text{usa}) \wedge \text{vende}(\text{west}, Z, m_1)$
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(8)** $\text{inimigo}(\text{nono}, \text{usa})$
 - Unifica com o objetivo e $\theta = \{Z/\text{nono}\}$
 - Adiciona premissa ao objetivo

Caso Coronel West

- Qual o novo objetivo?
 - $vende(west, nono, m_1)$
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(4)** $possui(nono, X) \wedge míssil(X) \rightarrow vende(west, nono, X)$
 - Unifica com o objetivo e $\theta = \{X/m_1\}$
 - Adiciona premissa ao objetivo

Caso Coronel West

- Qual o novo objetivo?
 - $\text{possui}(\text{nono}, m_1) \wedge \text{míssil}(m_1)$
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(2)** $\text{possui}(\text{nono}, m_1)$
 - Unifica com o objetivo
 - Adiciona premissa ao objetivo

Caso Coronel West

- Qual o novo objetivo?
 - *míssil*(m_1)
- Existe cláusula com um literal positivo que podemos unificar com o objetivo?
 - **(3)** *missil*(m_1)
 - Unifica com o objetivo
 - Adiciona premissa ao objetivo

Caso Coronel West

- Qual o novo objetivo?
 - Lista de objetivos vazia!
- Objetivo encontrado com sucesso!!!
 - Portanto West é criminoso!
- Caso contrário, não se pode afirmar que West é criminoso.

Encadeamento Regressivo

- **Desvantagem:** repetição de estados, podendo levar a ciclos infinitos.
- Exemplo:
 - $\text{caminho}(X, Y) \wedge \text{ligacao}(Y, Z) \rightarrow \text{caminho}(X, Z)$
- Prolog soluciona com memorização e retrocesso.
 - A ser mostrado mais adiante

SP com Encadeamento Regressivo

- Os grafos de execução e solução dependem dos dados iniciais e estratégia de resolução de conflito utilizada.
- Tenta provar cada objetivo uma **única vez**, **respeitando critério de resolução de conflito**.
 - Se a regra prova um objetivo já provado, ela é ignorada.

Exemplo1: SP com BC usando BP

► Exemplo:

1. $p, q \Rightarrow o$

2. $r, s \Rightarrow p$

3. $w, r \Rightarrow p$

4. $t, u \Rightarrow q$

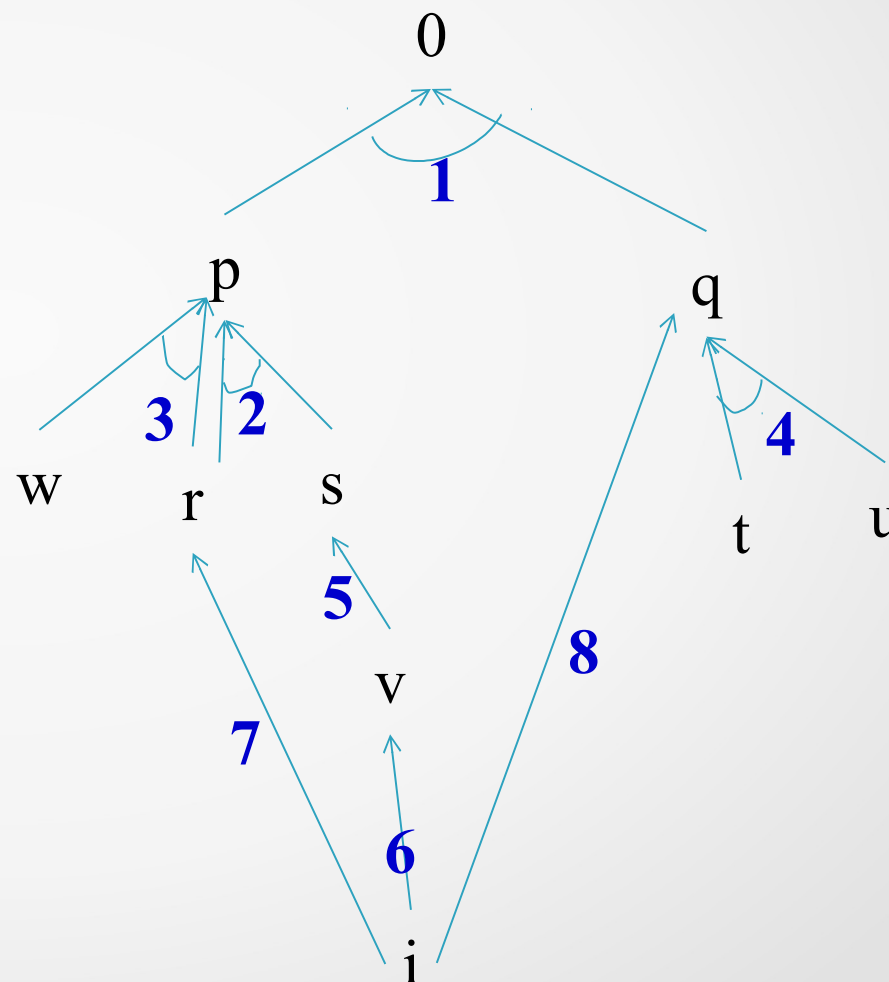
5. $v \Rightarrow s$

6. $i \Rightarrow v$

7. $i \Rightarrow r$

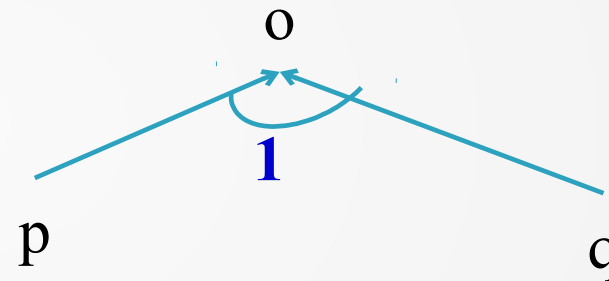
8. $i \Rightarrow q$

- Resolução de Conflito:
regra menos usada
até aquele ponto



Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$

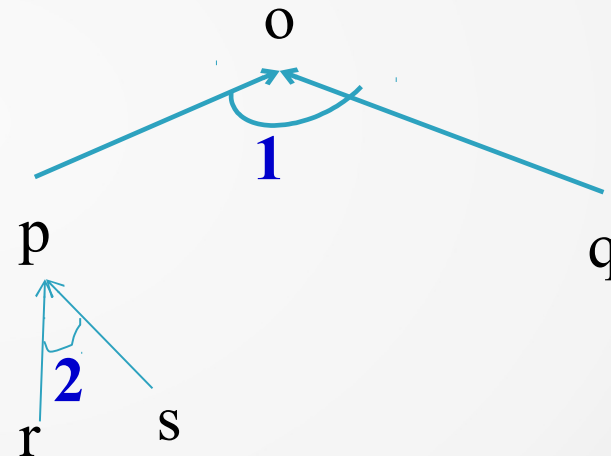


Dado inicial: i, t e u Objetivo: o

Resolução de conflito: regra menos usada até o momento

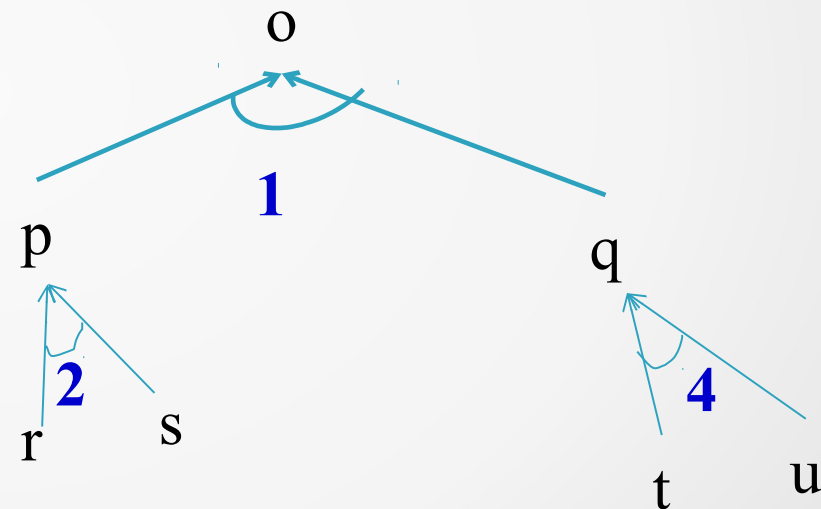
Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2,3,4,8	2

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$



Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2,3,4,8	2
2	o, p, q, r, s	4,5,7,8	4

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$



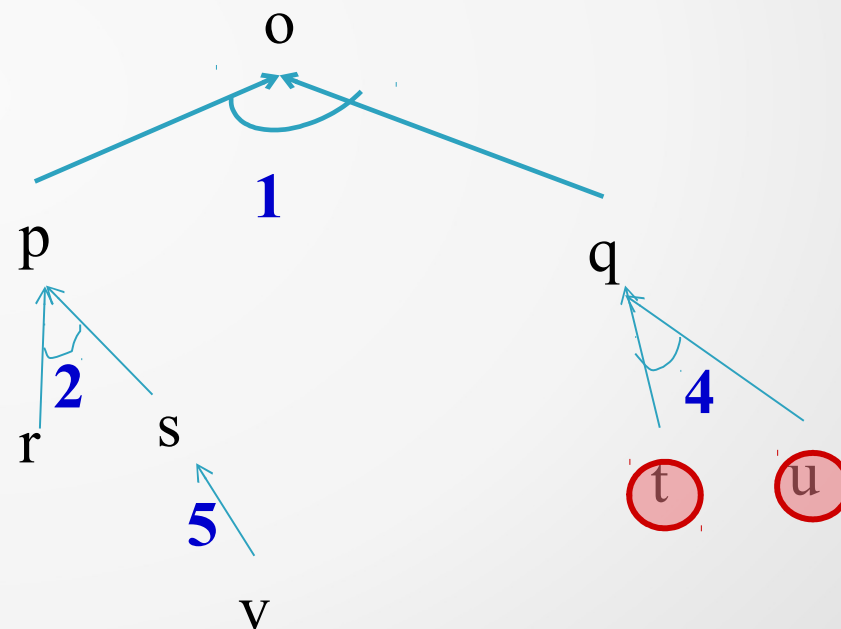
Observe que, como o objetivo p foi provado na iteração anterior, a regra 3 não é disparada mais. Portanto, a 4 é escolhida.

Objetivo: o

Dados iniciais : i, t, u

Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
2	o, p, q, r, s	4,5,7,8	4
3	o, p, q, r, s, t, u	5,7	5

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$



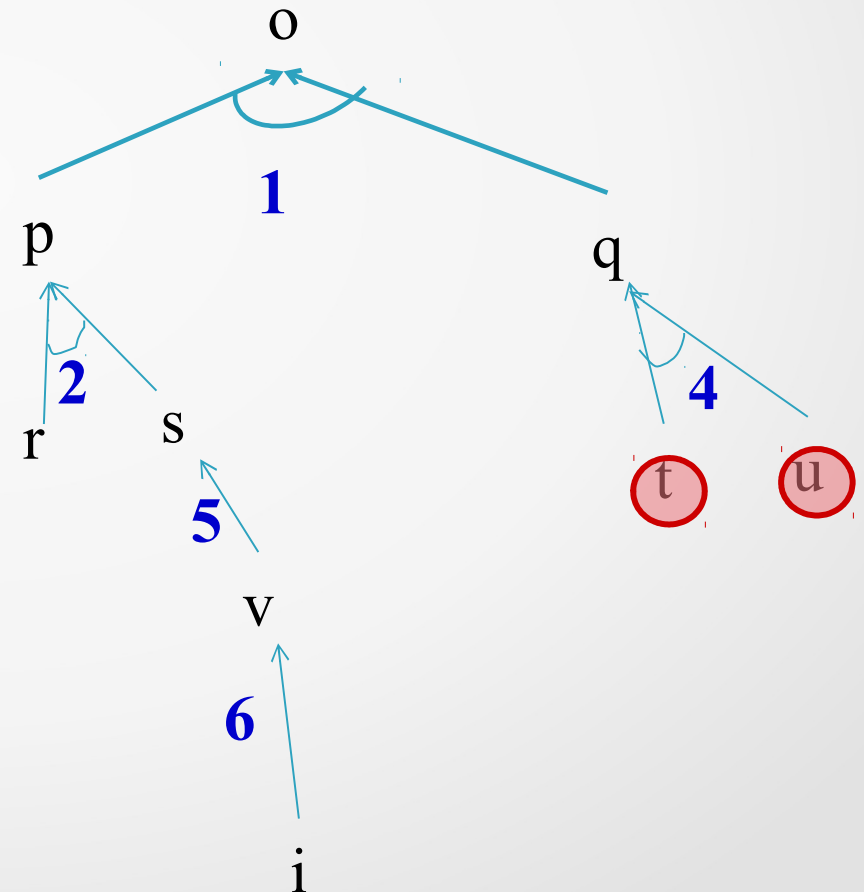
Observe que, como o objetivo q foi provado e, portanto, a regra 8 não é mais habilitada, assim como a regra 3.

Objetivo: o

Dados iniciais : i, t, u

Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
3	o, p, q, r, s, t, u	5, 7	5
4	o, p, q, r, s, t, u, v	6, 7	6

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$



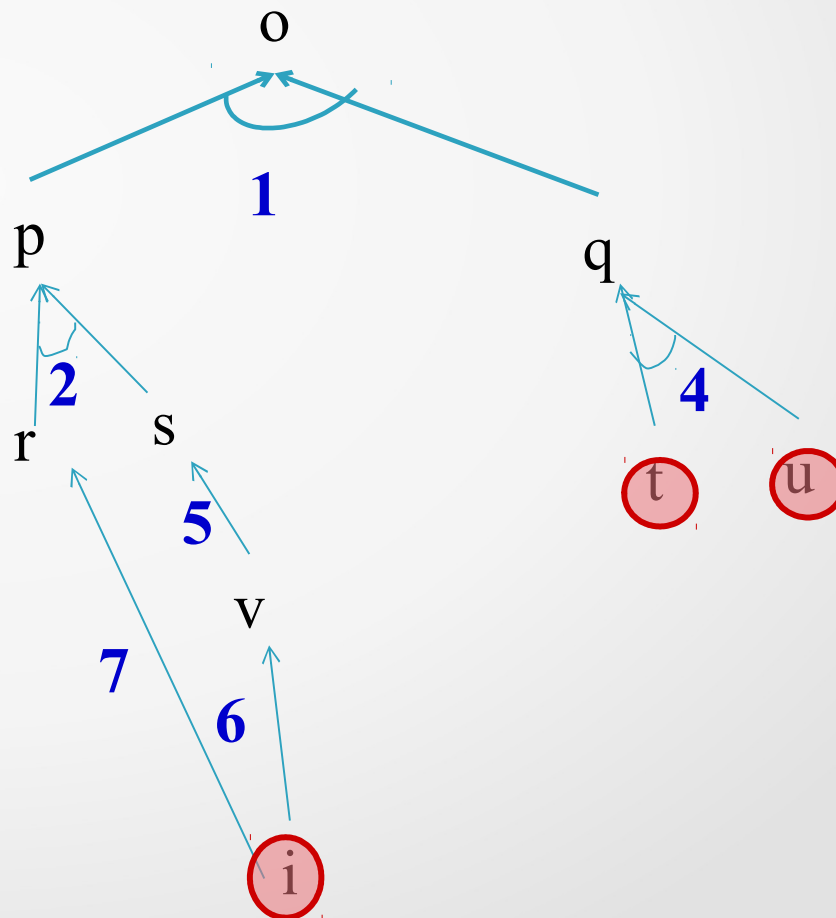
Objetivo: o

Dados iniciais : i, t, u

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
4	o, p, q, r, s, t, u, v	6, 7	6
5	o, p, q, r, s, t, u, v, i	7	7

1. $p, q \Rightarrow o$
2. $r, s \Rightarrow p$
3. $w, r \Rightarrow p$
4. $t, u \Rightarrow q$
5. $v \Rightarrow s$
6. $i \Rightarrow v$
7. $i \Rightarrow r$
8. $i \Rightarrow q$

Observe que o uso desta técnica evitou os disparos desnecessários das regras 3 e 8.



Sistemas de Produção com encadeamento para trás (Backward chaining)

Objetivo: o Dados iniciais : i, t, u

Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2,3,4,8	2
2	o, p, q, r, s	4,5,7,8	4
3	o, p, q, r, s, t, u	5,7	5
4	o, p, q, r, s, t, u, v	6,7	6
5	o, p, q, r, s, t, u, v, i	7	7
6	o, p, q, r, s, t, u, v, i		Pare

Retrocesso em BC usando BP

- ▶ No exemplo anterior, o SP encontrou uma solução evitando o disparo de duas regras desnecessárias.
- ▶ Contudo, ele poderia entrar em uma situação onde ele não conseguiria mais desenvolver a busca.
- ▶ Nesse caso, é necessário **retroceder** a algum ponto em que havia alguma regra para ser aplicada e não foi, e retomar a busca.

Exemplo2: SP com BC

► Exemplo:

1. $p, q \Rightarrow o$

2. $w, r \Rightarrow p$

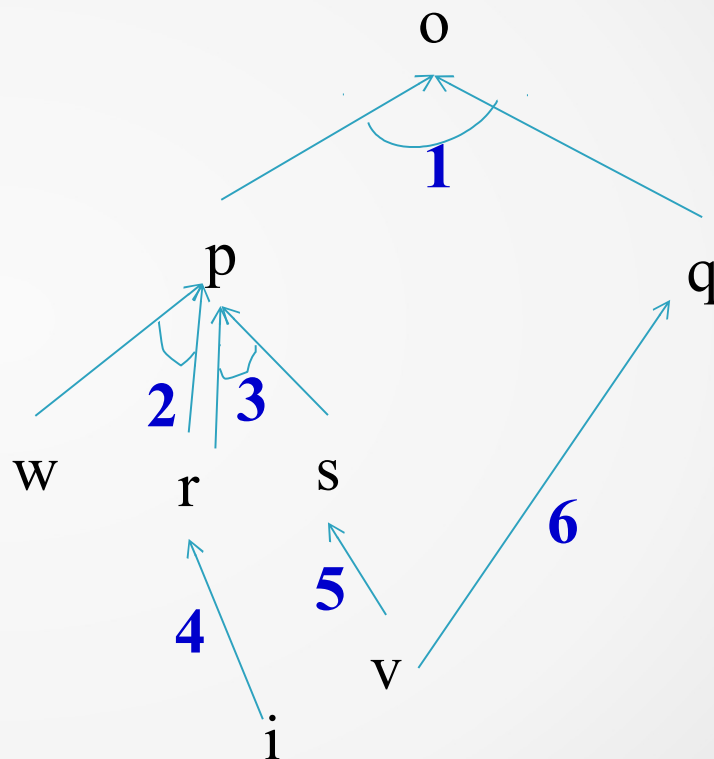
3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

5. $v \Rightarrow s$

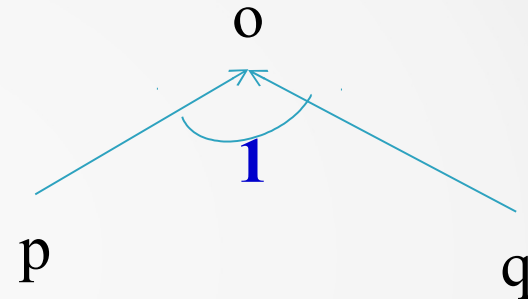
6. $v \Rightarrow q$

- Resolução de Conflito: regra menos usada até aquele ponto



Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1

1. $p, q \Rightarrow o$
2. $w, r \Rightarrow p$
3. $r, s \Rightarrow p$
4. $i \Rightarrow r$
5. $v \Rightarrow s$
6. $v \Rightarrow q$

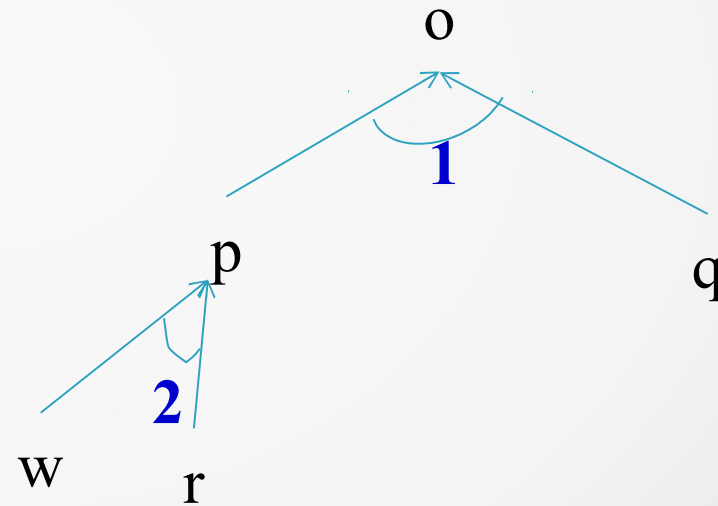


Dado inicial: i, v Objetivo: o

Resolução de conflito: regra menos usada até o momento

Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, <u>p</u> , <u>q</u>	2, 3, 6	2

1. $p, q \Rightarrow o$
2. $w, r \Rightarrow p$
3. $r, s \Rightarrow p$
4. $i \Rightarrow r$
5. $v \Rightarrow s$
6. $v \Rightarrow q$



Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	o	1	1
1	o, p, q	2, 3, 6	2
2	o, p, q, w, r	4, 6	4

1. $p, q \Rightarrow o$

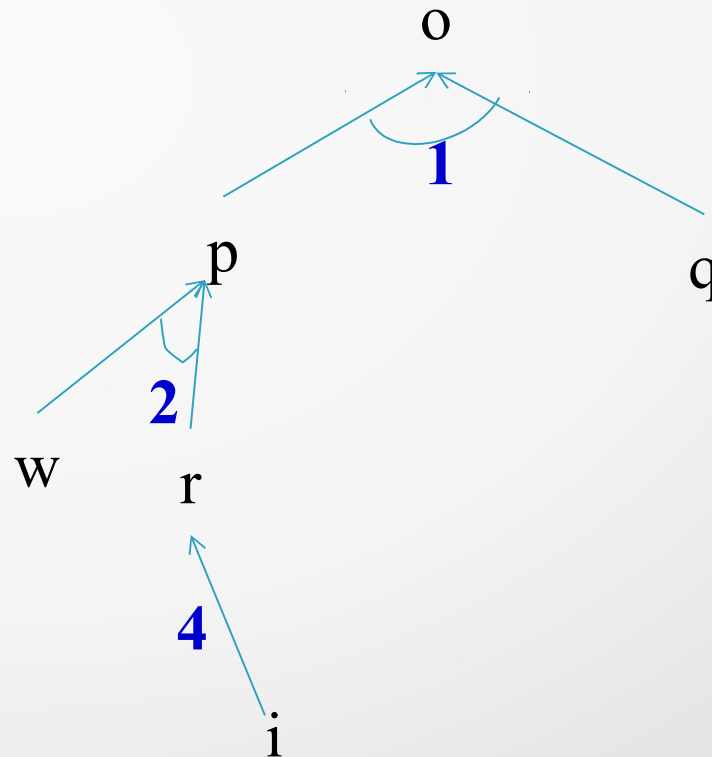
2. $w, r \Rightarrow p$

3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

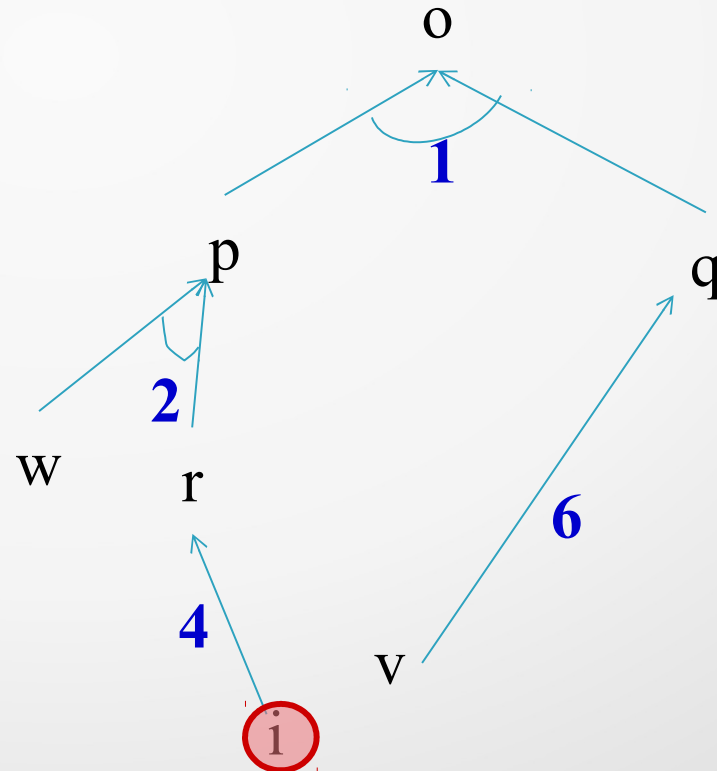
5. $v \Rightarrow s$

6. $v \Rightarrow q$



Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2, 3, 6	2
2	o, p, q, w, r	4, 6	4
3	o, p, q, w, r, i	6	6

1. $p, q \Rightarrow o$
2. $w, r \Rightarrow p$
3. $r, s \Rightarrow p$
4. $i \Rightarrow r$
5. $v \Rightarrow s$
6. $v \Rightarrow q$



Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	o	1	1
1	o, p, q	2, 3, 6	2
2	o, p, q, w, r	4, 6	4
3	o, p, q, w, r, i	6	6
4	o, p, q, w, r, i, v	NENHUMA o	--

1. $p, q \Rightarrow o$

2. $w, r \Rightarrow p$

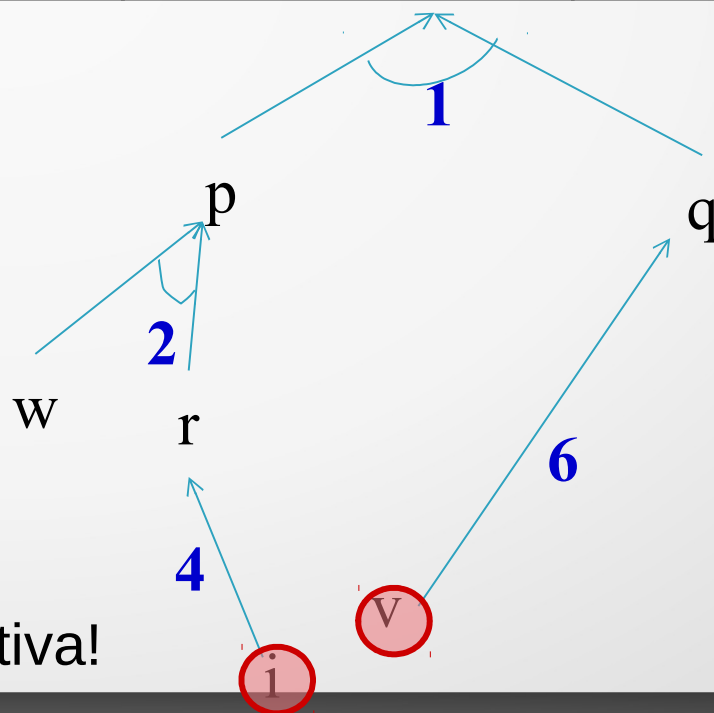
3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

5. $v \Rightarrow s$

6. $v \Rightarrow q$

Não há regra habilitada!
Necessário retroceder até alternativa!



Retrocesso

- ▶ Neste ponto, nenhuma regra está habilitada e a solução não foi encontrada.
- ▶ O processo retrocede ao ponto imediatamente anterior em que existe regra alternativa (iteração 2) e avança novamente, escolhendo outra regra.

Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	o	1	1
1	o, p, q	2, 3, 6	2
2	o, p, q, w, r	4, 6	4
3	o, p, q, w, r, i	6	6
4	o, p, q, w , r, i , v	NENHUMA o	--

1. $p, q \Rightarrow o$

2. $w, r \Rightarrow p$

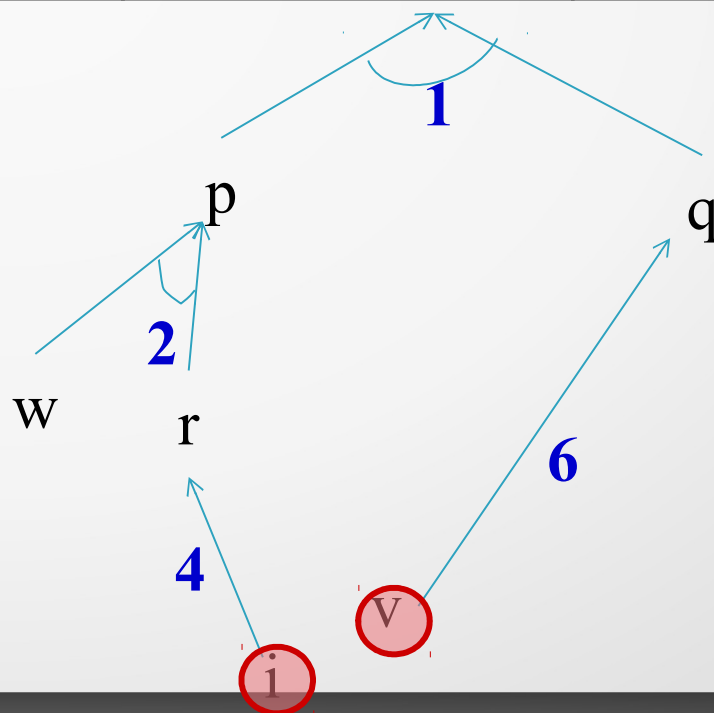
3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

5. $v \Rightarrow s$

6. $v \Rightarrow q$

É preciso voltar até a iteração 2 e escolher a regra 6!



Iteração	Memória de Trabalho	Conjunto de conflito	Disparo
0	o	1	1
1	o, p, q	2, 3, 6	2
2	o, p, q, w, r	4, 6	6
3	o, p, q, w, r, v	4	4
4	o, p, q, w, r, v, i	NENHUMA	--

1. $p, q \Rightarrow o$

2. $w, r \Rightarrow p$

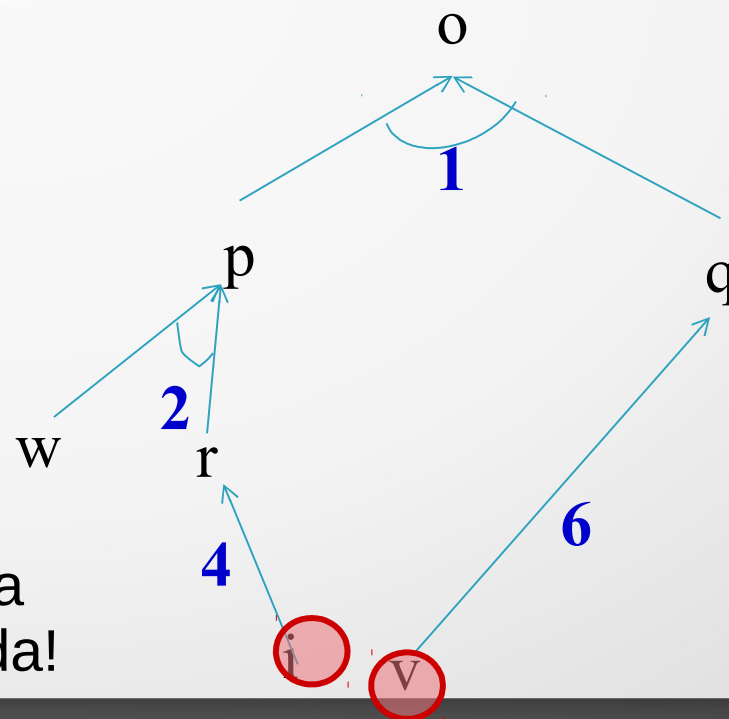
3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

5. $v \Rightarrow s$

6. $v \Rightarrow q$

Contudo, ao desenvolver, a busca ainda fica sem nenhuma habilitada!



Retrocesso

- ▶ Neste ponto, novamente nenhuma regra está habilitada e a solução não foi encontrada.
- ▶ O processo retrocede ao ponto imediatamente anterior em que existe regra alternativa (iteração 1) e avança novamente, escolhendo outra regra.

Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2, 3, 6	3

1. $p, q \Rightarrow o$

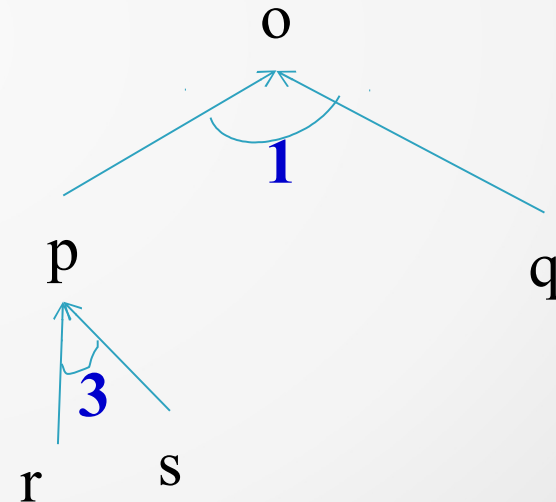
2. $w, r \Rightarrow p$

3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

5. $v \Rightarrow s$

6. $v \Rightarrow q$



É preciso voltar até a iteração 1 e escolher a regra 3!

Iteração	Memória de Trabalho	Conjunto de conflito	Disparco
0	o	1	1
1	o, p, q	2, 3, 6	3
2	o, p, q, r, s	4, 5, 6	4

1. $p, q \Rightarrow o$

2. $w, r \Rightarrow p$

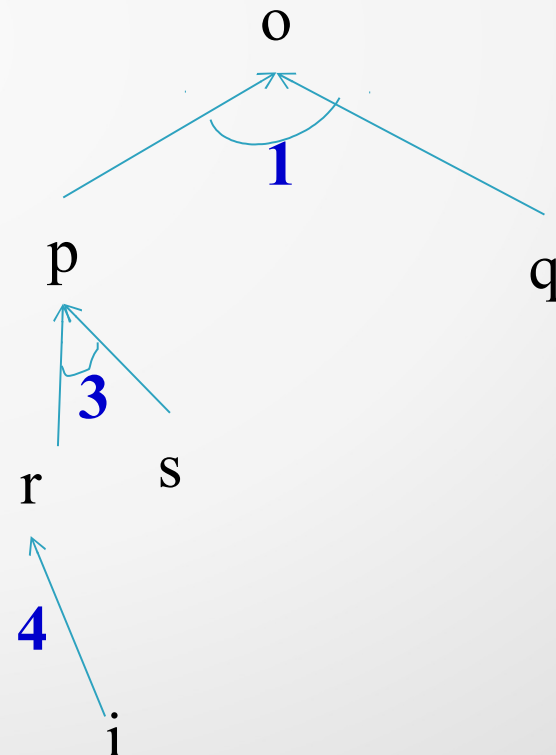
3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

5. $v \Rightarrow s$

6. $v \Rightarrow q$

Agora a busca segue por um novo caminho! Regra 2 não está habilitada!



Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2, 3, 6	3
2	o, p, q, r, s	4, 5, 6	4
3	o, p, q , r, s , i	5, 6	5

1. $p, q \Rightarrow o$

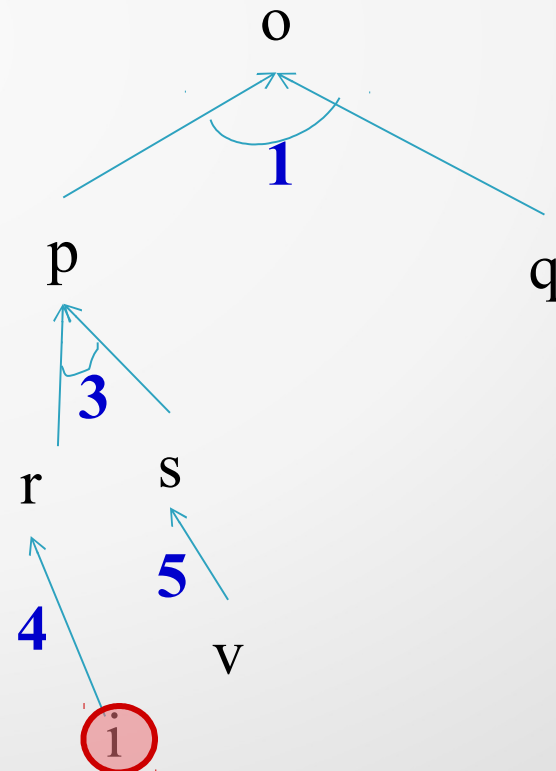
2. $w, r \Rightarrow p$

3. $r, s \Rightarrow p$

4. $i \Rightarrow r$

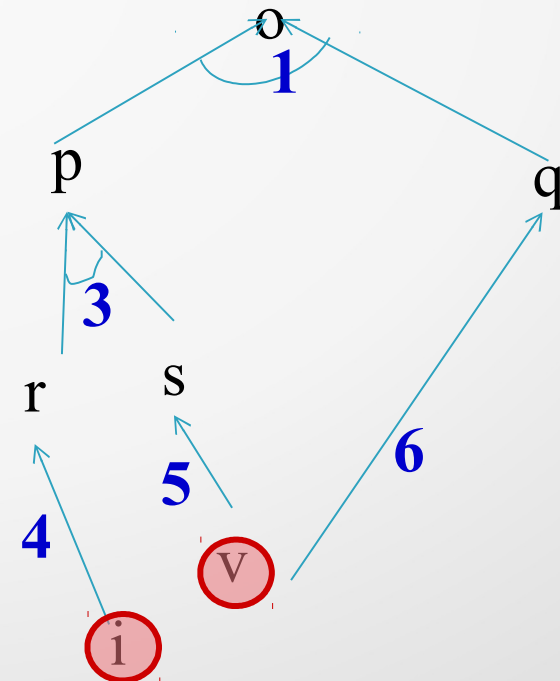
5. $v \Rightarrow s$

6. $v \Rightarrow q$



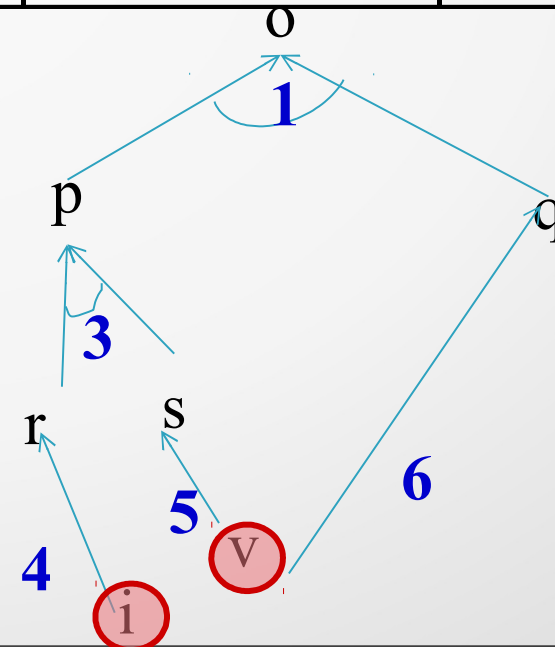
Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2, 3, 6	3
2	o, p, q, r, s	4, 5, 6	4
3	o, p, q, r, s, i	5, 6	5
4	o, p, q, r, s, i, v	6	6

1. $p, q \Rightarrow o$
2. $w, r \Rightarrow p$
3. $r, s \Rightarrow p$
4. $i \Rightarrow r$
5. $v \Rightarrow s$
6. $v \Rightarrow q$

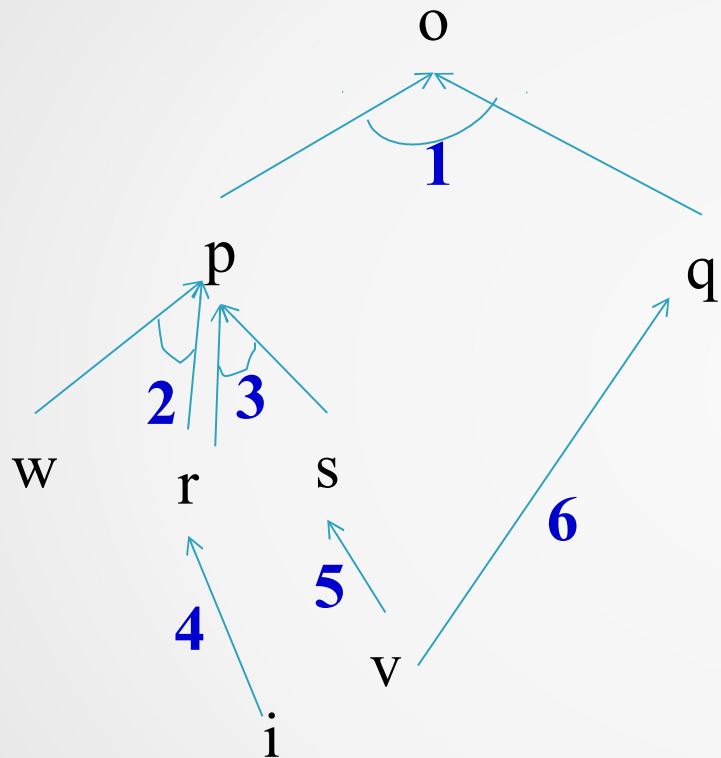


Iteração	Memória de Trabalho	Conjunto de conflito	Disparc
0	o	1	1
1	o, p, q	2, 3, 6	3
2	o, p, q, r, s	4, 5, 6	4
3	o, p, q, r, s. i	5, 6	5
4	o, p, q, r, s. i, v	6	6
5	o, p, q, r, s. i, v	--	Pare

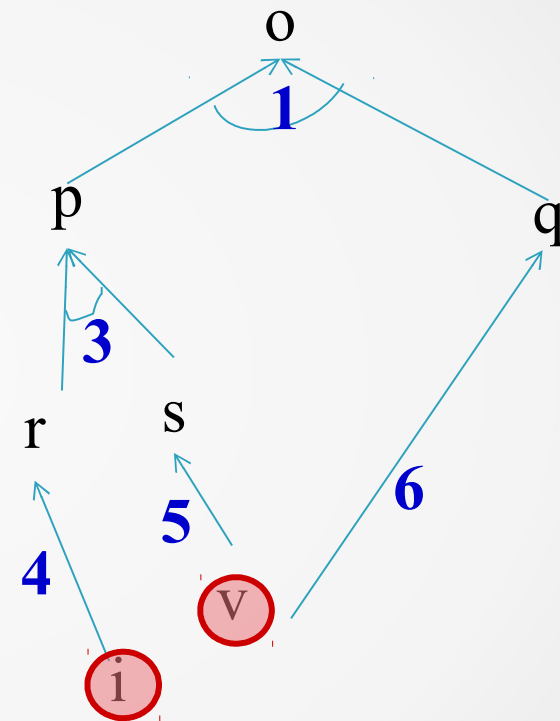
1. $p, q \Rightarrow o$
2. $w, r \Rightarrow p$
3. $r, s \Rightarrow p$
4. $i \Rightarrow r$
5. $v \Rightarrow s$
6. $v \Rightarrow q$



SP com BC usando BP



Grafo completo



Grafo de execução

Resolução

- Um terceiro método de inferência é conhecido como Resolução.
- Consiste em resolver cláusulas com literais proposicionais complementares.
 - Literais são complementares se um é a negação do outro.
 - Todas as cláusulas devem ser disjunções de literais dos quais no máximo um é positivo
 - Esse tipo de cláusula é chamada de cláusula de Horn.

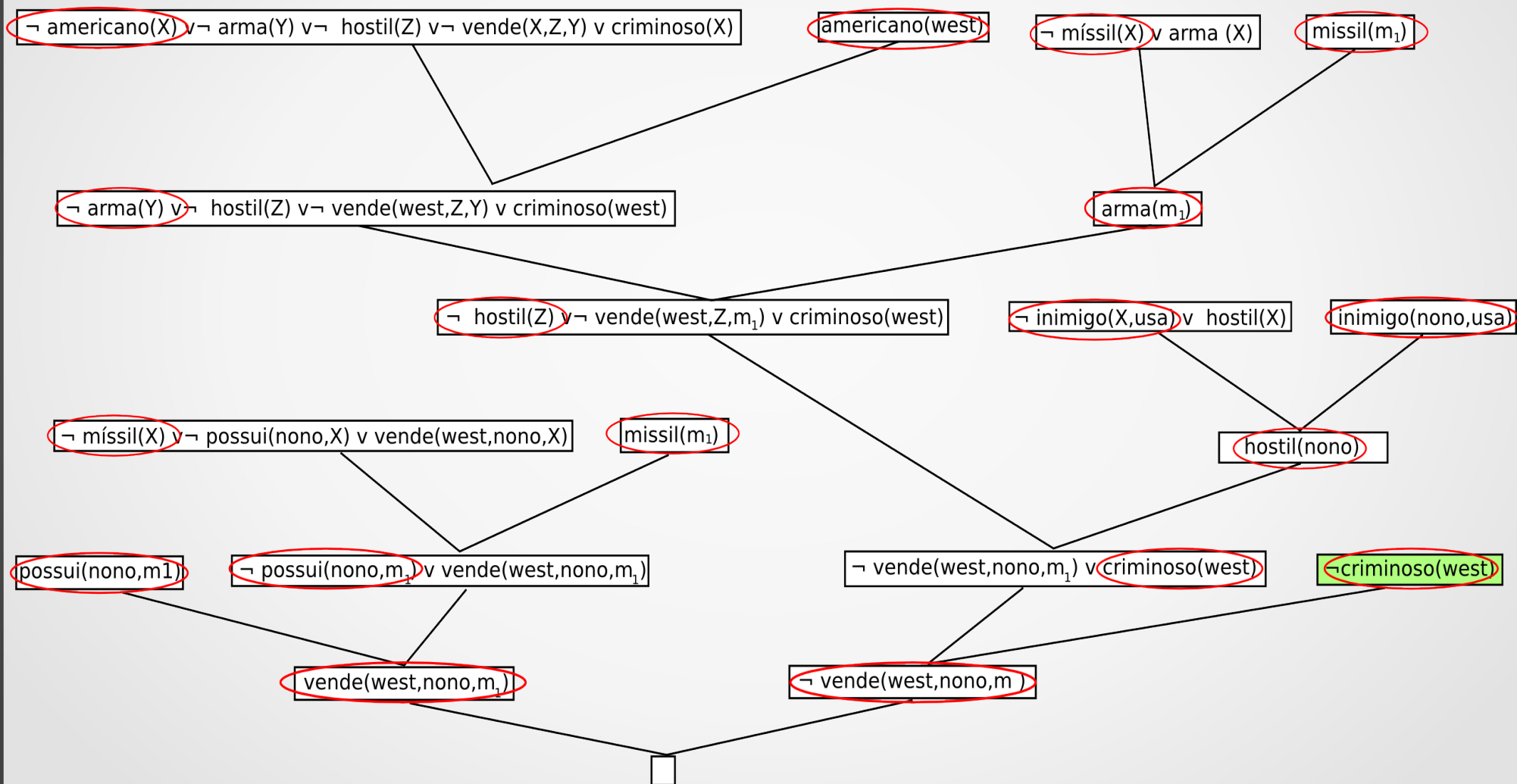
Coronel West

- Como resolver o caso do coronel usando resolução
 - $\text{americano}(X) \wedge \text{arma}(Y) \wedge \text{hostil}(Z) \wedge \text{vende}(X,Z,Y) \rightarrow \text{criminoso}(X)$
- Na FNC se torna:
 - $\neg \text{americano}(X) \vee \neg \text{arma}(Y) \vee \neg \text{hostil}(Z) \vee \neg \text{vende}(X,Z,Y) \vee \text{criminoso}(X)$

Resolução do Coronel

- As sentenças na FNC tornam-se:
 - $\neg \text{míssil}(X) \vee \neg \text{possui}(\text{nono}, X) \vee \text{vende}(\text{west}, \text{nono}, X)$
 - $\neg \text{inimigo}(X, \text{usa}) \vee \text{hostil}(X)$
 - $\neg \text{míssil}(X) \vee \text{Arma}(X)$
 - $\text{possui}(\text{nono}, m_1)$
 - $\text{míssil}(m_1)$
 - $\text{americano}(\text{west})$
 - $\text{inimigo}(\text{nono}, \text{usa})$

Solução



Para saber mais detalhes, consulte o livro do Russel e Norvig indicado na bibliografia básica.

Em resumo

- Vimos que SP podem utilizar mecanismos de inferência para encontrar ações
- Dois mecanismos foram apresentados em aula
 - Encadeamento progressivo ou para frente
 - Orientado a dados
 - Encadeamento regressivo ou para trás
 - Orientado a objetivo
 - Que pode usar BP para aumentar eficiência
 - Mas pode requerer o uso de retrocesso.
- Resolução é um terceiro tipo de SP e é utilizado por resolvedores de problemas como o Prolog.