



INTELIGÊNCIA ARTIFICIAL
INFERÊNCIA ASCENDENTE E DESCENDENTE

Aldisio Medeiros

aldisiog@gmail.com

Carlos Adailton Rodrigues

adtn7000@gmail.com

Roteiro

1. Conceitos Iniciais
2. Cláusulas de Horn
 - 2.1 Definições
 - 2.2 Razões para o Uso
3. Algoritmos para Inferência com Cláusulas de Horn
 - 2.1 Encadeamento para a frente
 - Características
 - Pseudocódigo
 - 2.2 Encadeamento para trás
 - Características
 - Pseudocódigo
4. Dúvidas

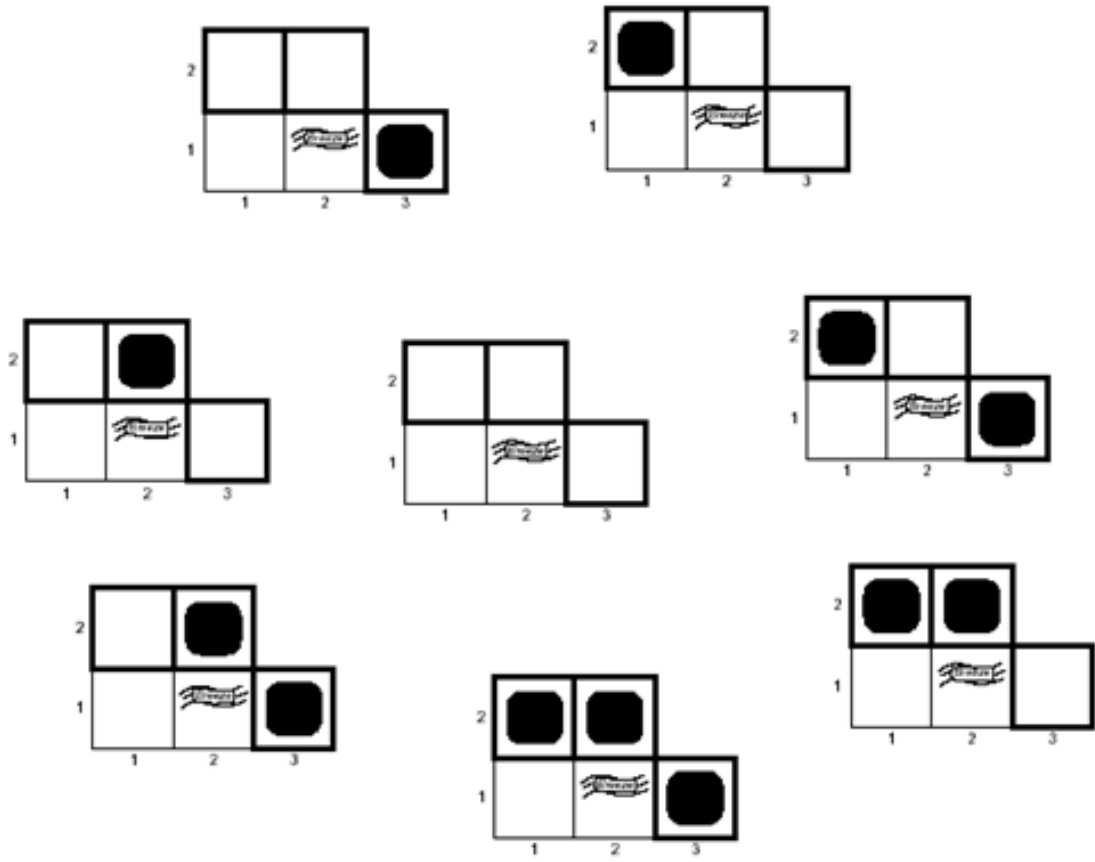
Conceitos Iniciais

- Bases de conhecimento
 - Pode ser definida como uma declaração;
 - Deve manter sua **validade** para todos os modelos válidos do mundo
 - É composta por um **conjunto de setenças**;

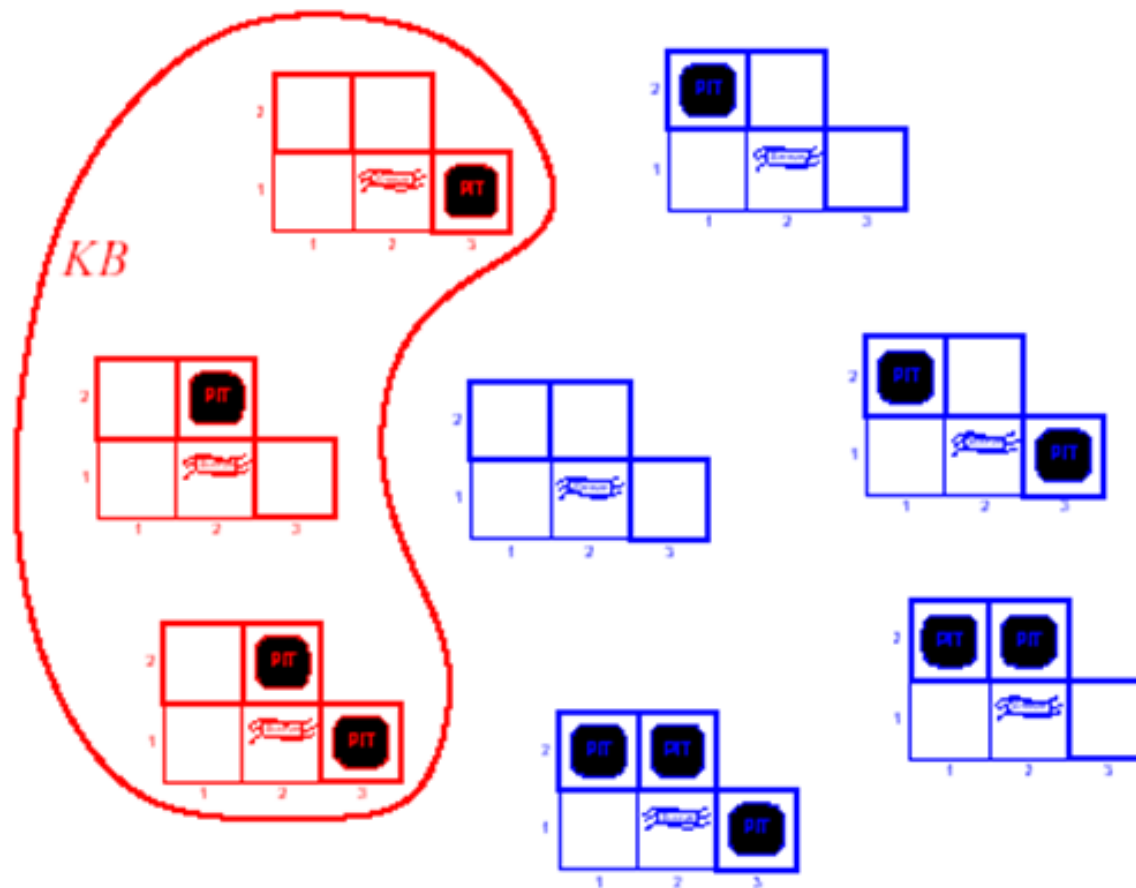
$$BC \models \alpha$$

- A sentença α deve ser válida no conjunto de modelos M
- As sentenças são expressas em uma **linguagem de representação de conhecimento**. Definem operações de TELL e ASK, por exemplo;
- Utilização do conhecimento prévio (BC) para a dedução de novos conhecimentos e a partir disso, tomar decisões. Ex: Mundo de Wumpus

Conceitos Iniciais

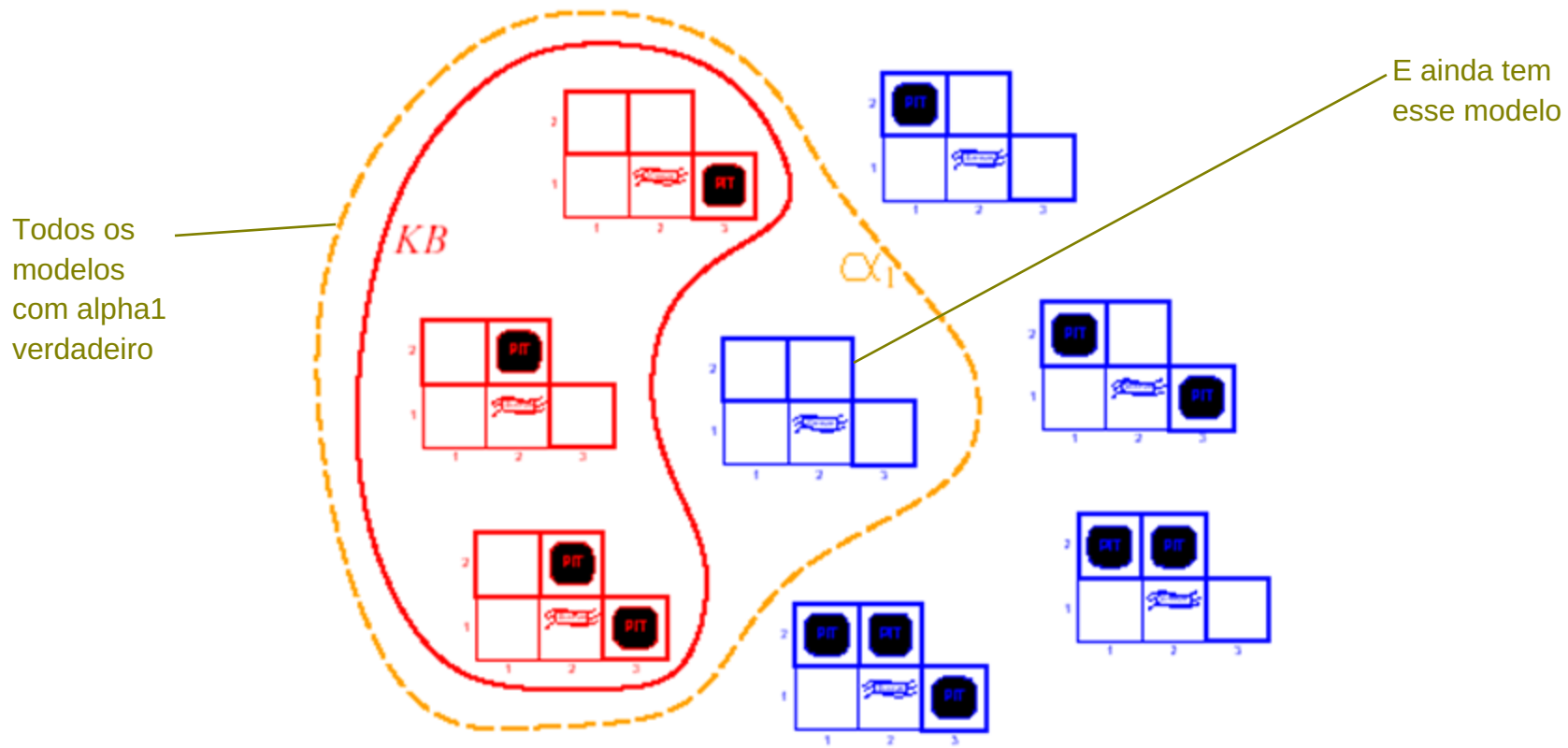


Conceitos Iniciais



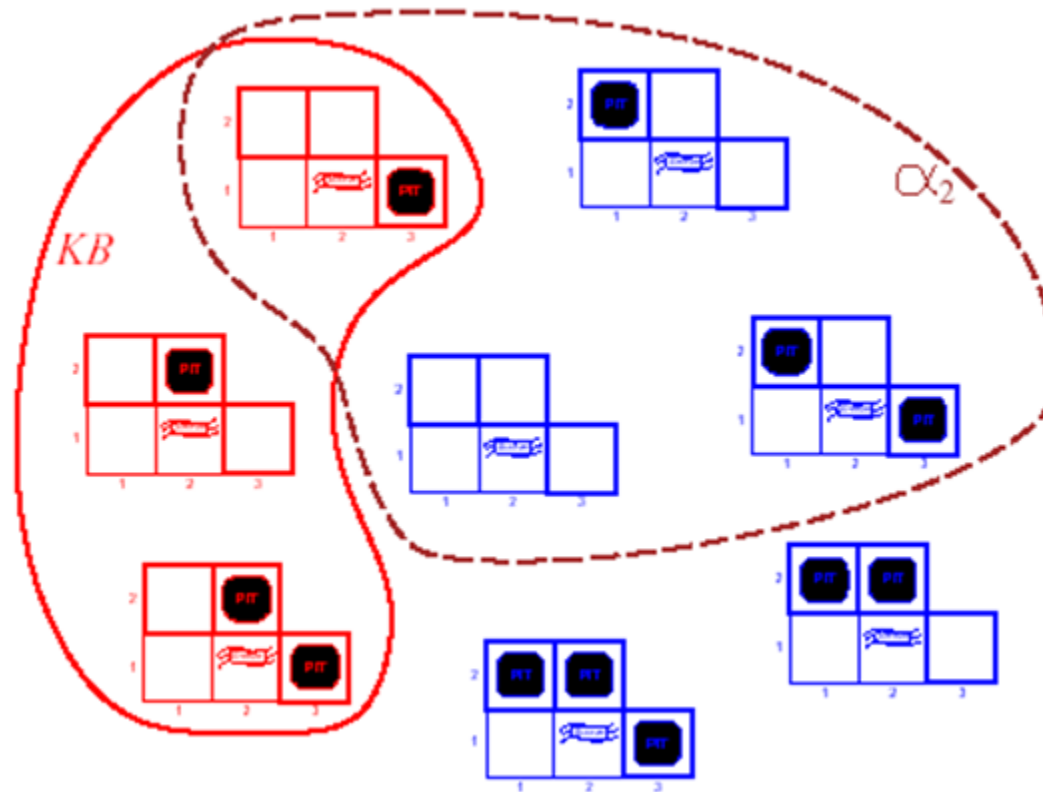
KB: Conhecimento + Percepções

Conceitos Iniciais



α_1 = “não existe nenhum poço em [1,2]” . Portanto $BC \models \alpha_1$ é válido!

Conceitos Iniciais



KB não é subconjunto de α_2

α_2 = “não existe nenhum poço em [2,2]”. Portanto $BC \models \alpha_2$ não é válido!

Conceitos Iniciais

Portanto, com os conhecimentos existentes na base de conhecimento a respeito do mundo, e a partir das percepções, conseguimos encontrar modelos onde a implicação lógica $BC \models \alpha$, é válida.

Conseguimos inferir que:

1) É seguro ir para [1, 2]. Onde $BC \models \alpha_1$ é válida.

2) Não temos certeza se é seguro ir para [2,2]. Onde $BC \models \alpha_2$ não é válida.

*“A busca pode **avançar** a partir da base de conhecimento inicial, aplicando regras de inferência para **derivar a sentença objetivo**, ou pode **retroceder** a partir da sentença objetivo tentando encontrar uma cadeia de regras de inferência que tenham origem na BC inicial”
(Russel e Norvig, 2014)*

Cláusulas de Horn - Definições

- Bases de conhecimento reais muitas vezes utilizam apenas cláusulas de uma espécie restrita.
- Cláusula de Horn é uma disjunção de literais dos quais no máximo um é positivo.
 - Ex.:
 - $(\neg L_{1,1} \vee \neg \text{Brisa} \vee B_{1,1})$ é cláusula de Horn
 - $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1})$ não é cláusula de Horn
- **Cláusulas definidas** são aquelas com exatamente um literal positivo.
- O literal positivo é chamado **cabeça** e os negativos formam o **corpo** da cláusula.
- **Restrição de integridade** é quando temos uma cláusula de Horn sem literais positivos, que pode ser escrita como uma implicação cuja conclusão é o literal Falso. Ex.:
 $(\neg W_{1,1} \vee \neg W_{1,2})$ é equivalente a $W_{1,1} \wedge W_{1,2} \Rightarrow \text{Falso}$

Cláusulas de Horn - Razões para o Uso

1. Toda cláusula de Horn pode ser escrita como uma implicação cuja premissa é uma conjunção de literais positivos e cuja conclusão é um único literal positivo.

Ex.:

$(\neg L_{1,1} \vee \neg \text{Brisa} \vee B_{1,1})$

pode ser escrita como:

$(L_{1,1} \wedge \text{Brisa}) \Rightarrow B_{1,1}$

2. A decisão de consequência lógica com cláusulas de Horn pode ser feita em tempo linear em relação ao tamanho da base de conhecimento.

Cláusulas de Horn - Razões para o Uso

3. A inferência pode ser feita através dos algoritmos de **encadeamento para frente** e **encadeamento para trás**. Ambos são naturais, pois possuem etapas de inferência fáceis para seres humanos.

Algoritmos para Inferência com Cláusulas de Horn

- Nos algoritmos a seguir, supomos por simplicidade que a base de conhecimento contém somente cláusulas definidas e nenhuma restrição de integridade.
- Bases de Conhecimento na forma de Horn;

Encadeamento pra Frente - EF

função CONSEQUÊNCIA-LOGICA-LP-EF?(BC, q) retorna verdadeiro ou falso

entradas: BC, um conjunto de cláusulas de Horn proposicionais

q, um símbolo proposicional

variáveis locais: contagem, deduzido, agenda

enquanto agenda não é vazia **faça**

p <- POP(agenda)

a menos que deduzido[p] **faça**

deduzido[p] <- verdadeiro

para cada cláusula de Horn c em cuja premissa p aparece **faça**

decrementar contagem[c]

se contagem[c] = 0 **então faça**

se CABEÇA[C] = q **então retornar** verdadeiro

EMPILHAR (CABEÇA[c], agenda)

retornar falso

Encadeamento pra Frente - EF

Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.

$$P \Rightarrow Q$$

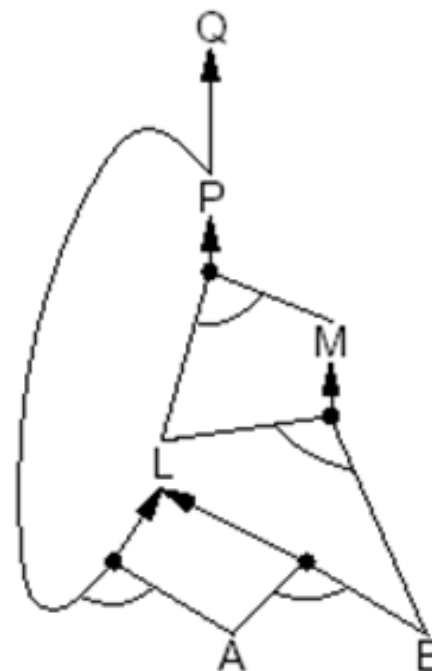
$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$


$$A \wedge B \Rightarrow L$$

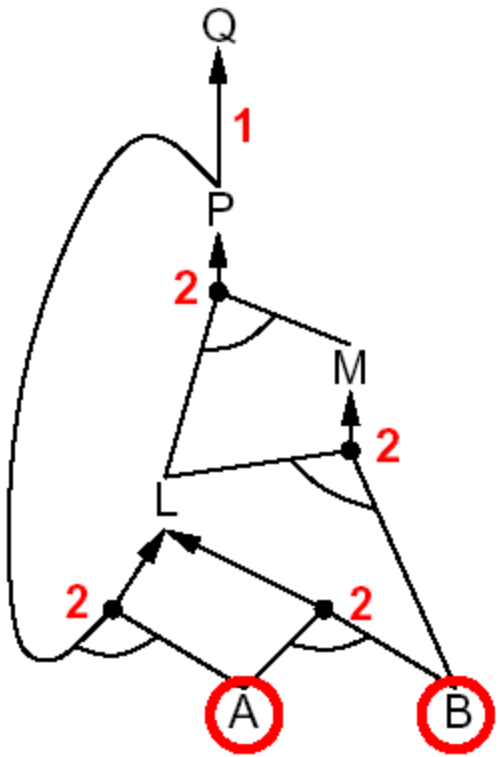
$$A$$

$$B$$


Encadeamento pra Frente - EF

Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.

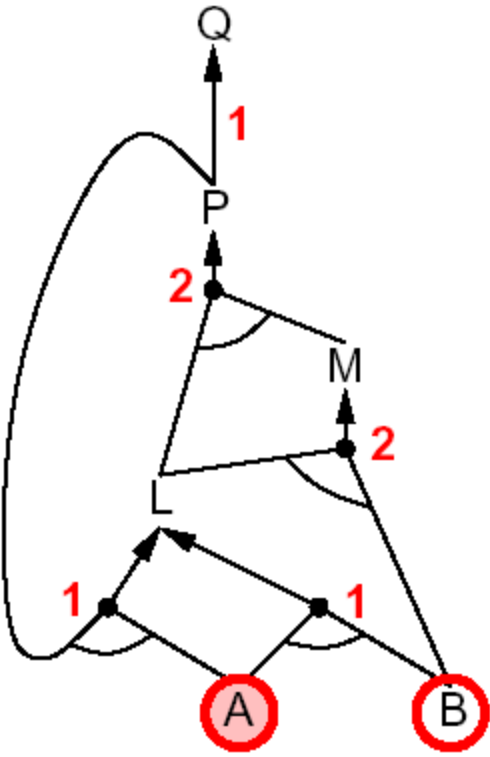
 é um fato ou algo que já já foi concluído



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento pra Frente - EF

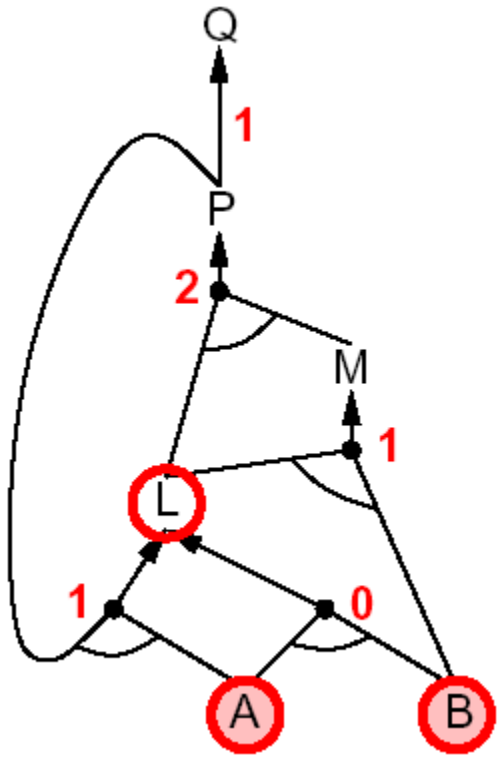
Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento pra Frente - EF

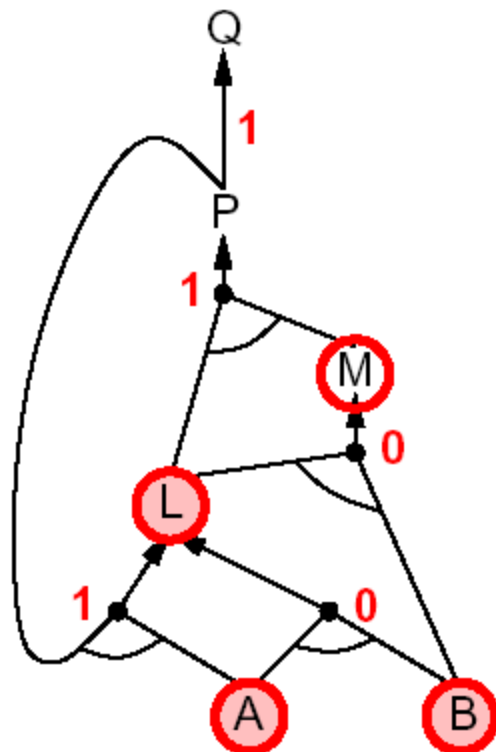
Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento pra Frente - EF

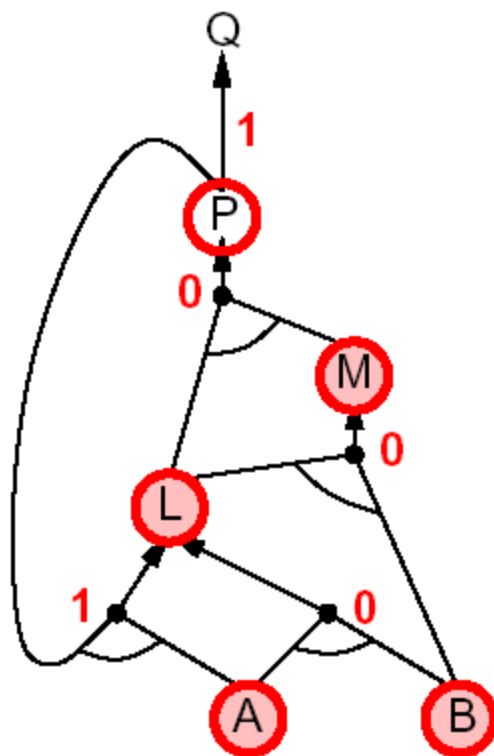
Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento pra Frente - EF

Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

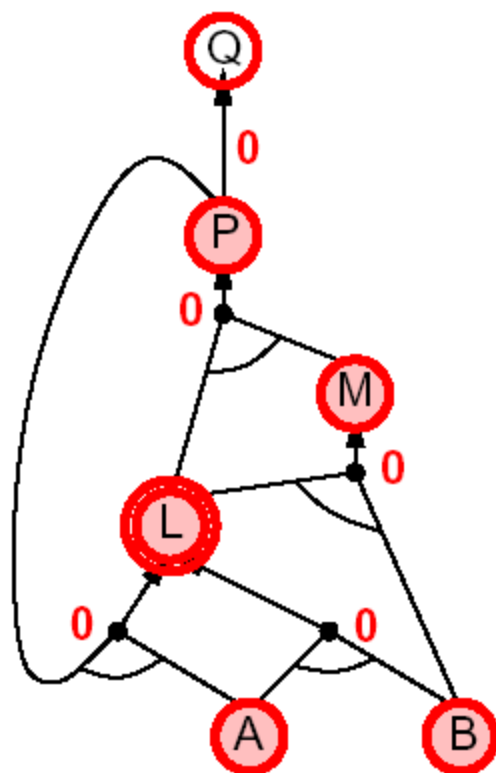
$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

Encadeamento pra Frente - EF

Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.



$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

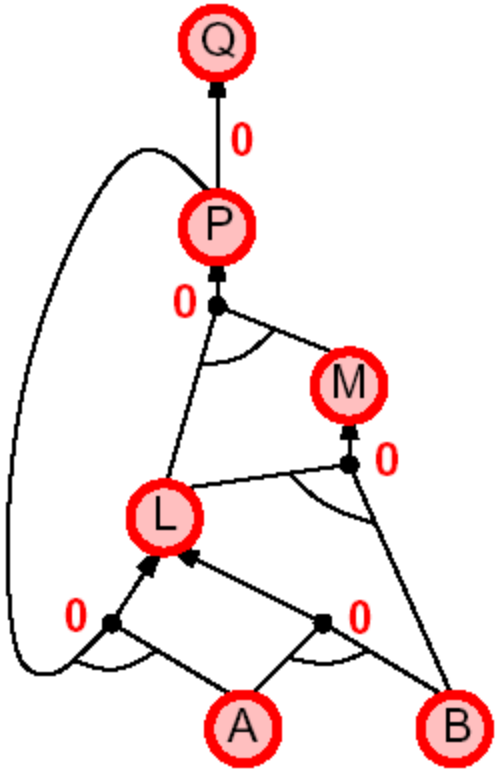
$$A \wedge B \Rightarrow L$$

A

B

Encadeamento pra Frente - EF

Idéia: descarte qualquer regra cujas premissas são satisfeitas na KB, adicione sua conclusão à KB, até que a regra procurada seja encontrada.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento pra Frente - EF

Características

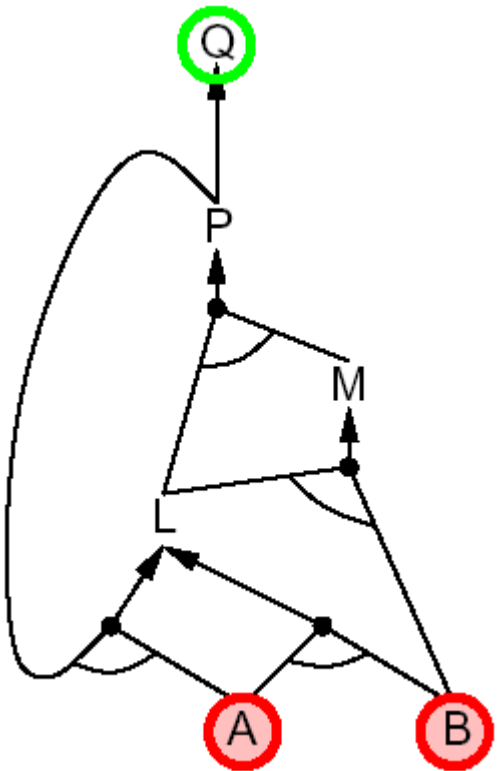
Raciocínio orientado a dados;

Consistente;

Completo;

Encadeamento para Trás

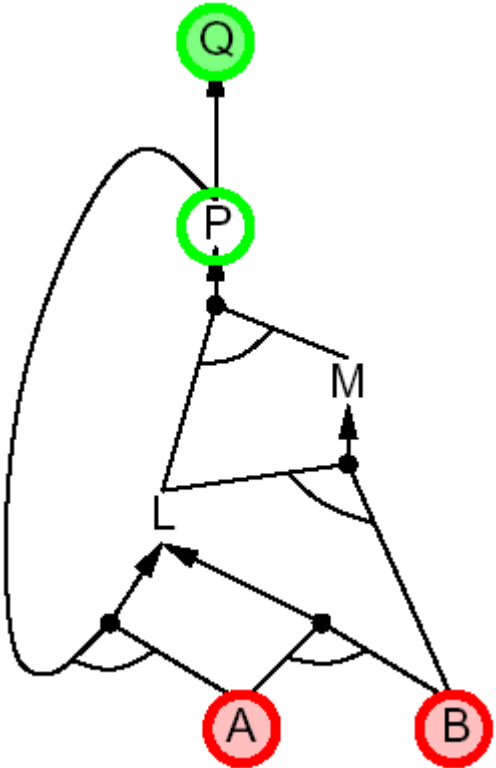
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- | |
|-------------------|
| $P \Rightarrow Q$ |
|-------------------|
- $L \wedge M \Rightarrow P$
 - $B \wedge L \Rightarrow M$
 - $A \wedge P \Rightarrow L$
 - $A \wedge B \Rightarrow L$
 - A
 - B

Encadeamento para Trás

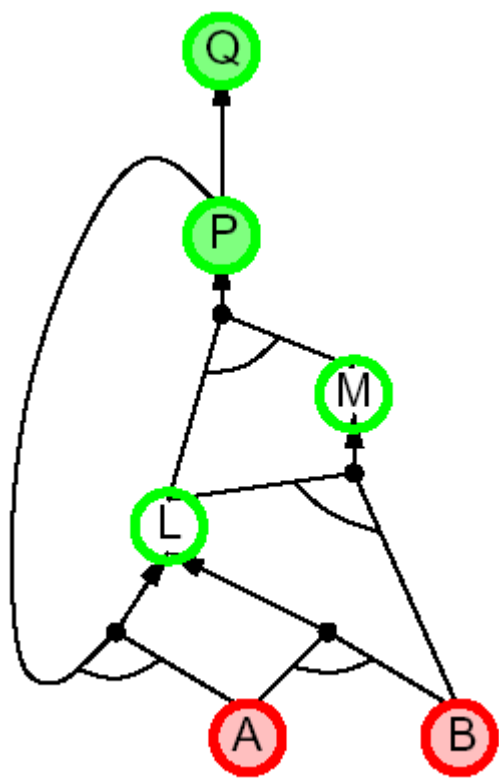
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

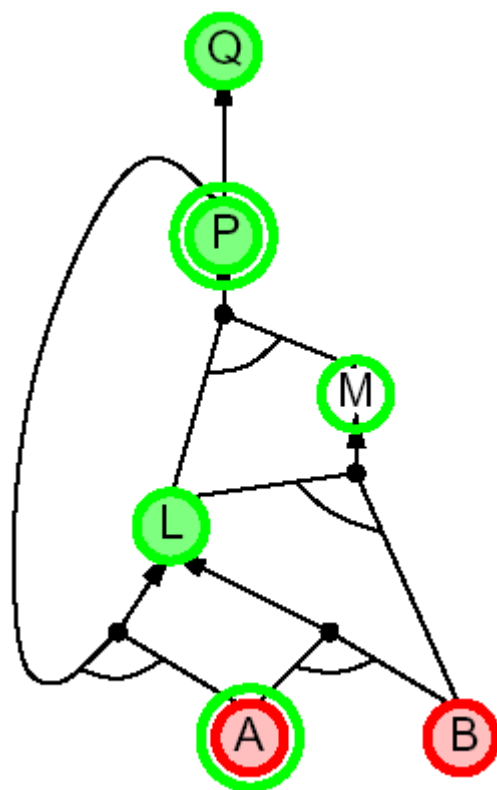
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

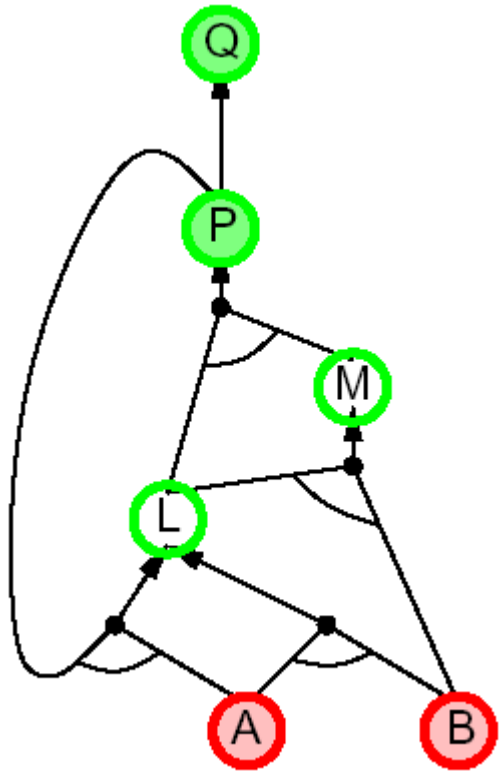
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



$P \Rightarrow Q$
 $L \wedge M \Rightarrow P$
 $B \wedge L \Rightarrow M$
 $A \wedge P \Rightarrow L$
 $A \wedge B \Rightarrow L$
 A
 B

Encadeamento para Trás

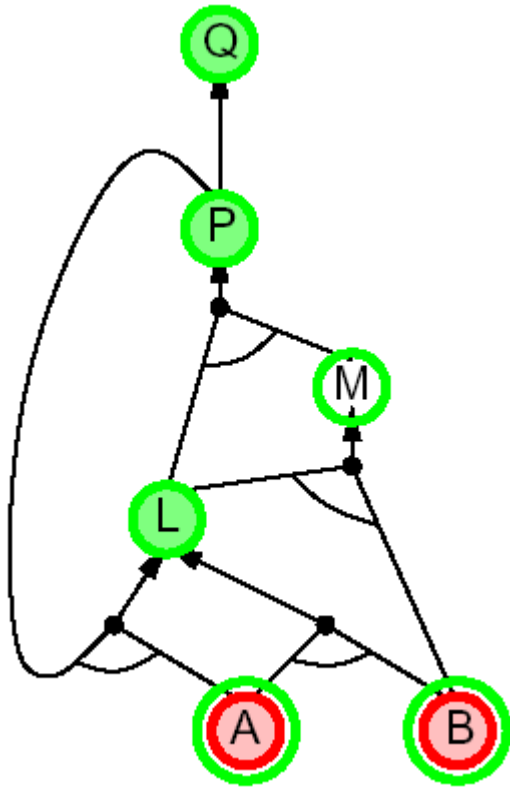
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

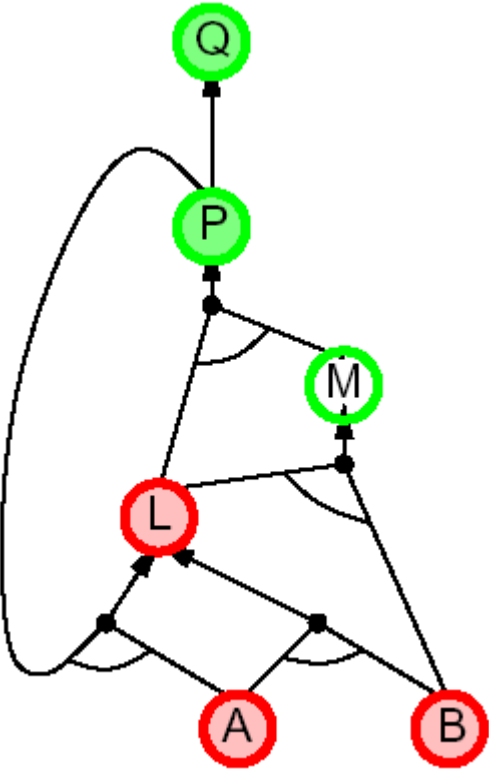
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

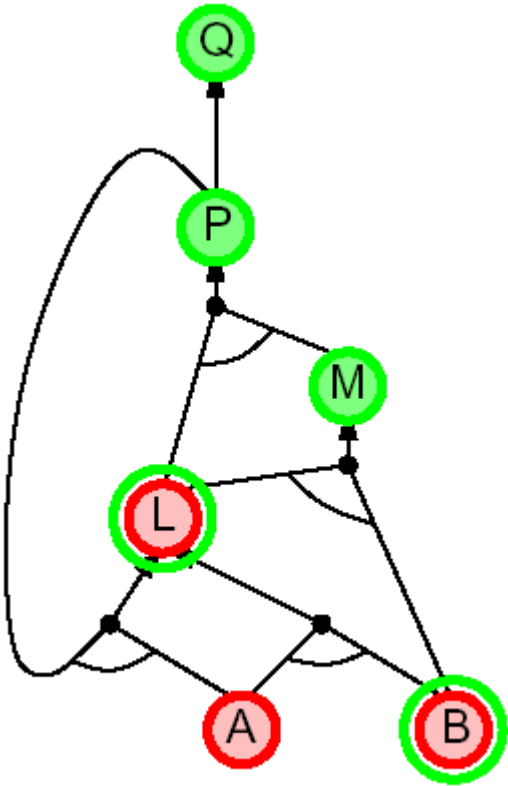
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

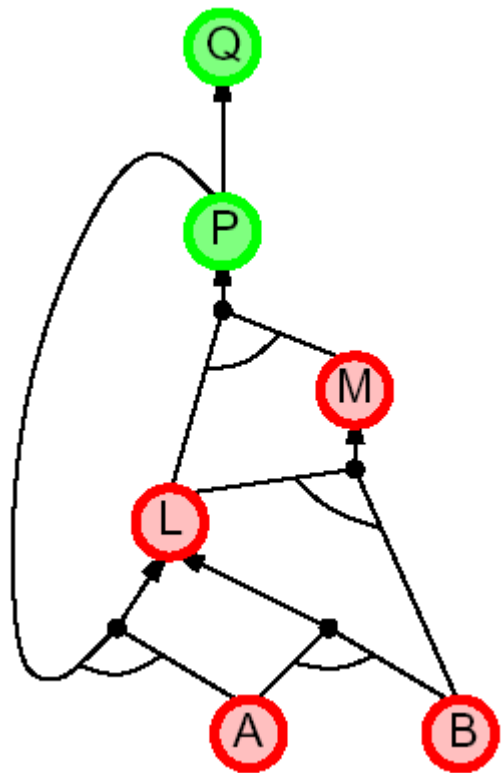
Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



$$P \Rightarrow Q$$

$L \wedge M \Rightarrow P$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

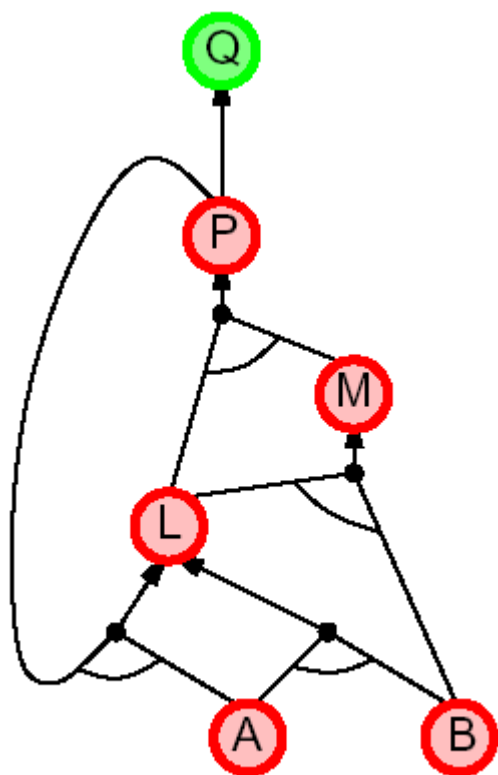
$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

Encadeamento para Trás

Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q .



$P \Rightarrow Q$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

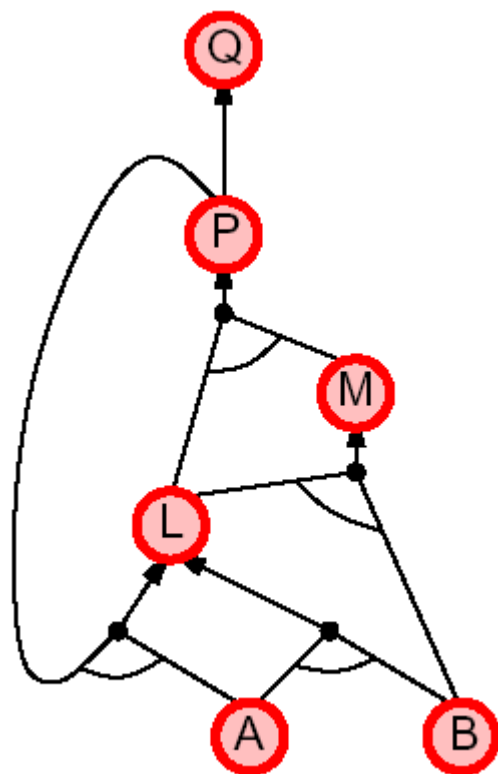
$$A \wedge B \Rightarrow L$$

$$A$$

$$B$$

Encadeamento para Trás

Idéia: para provar Q por BC, checar se Q já é conhecido, ou provar por BC todas as premissas de alguma regra, concluindo Q.



- $P \Rightarrow Q$
- $L \wedge M \Rightarrow P$
- $B \wedge L \Rightarrow M$
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$
- A
- B

Encadeamento para Trás

se Q corresponde a um fato básico **então**

Q é verdadeiro e provar (Q) tem sucesso

Gerar/Adicionar fatos

Guardar justificações

Encadeamento para Trás

senão se há um fato F que contraria Q **então**

Q é falso e provar (Q) falhou

SR <- conjunto de regras que concluem Q

repetir

R <- regra de SR

SC <- Conjunto de condições de R

Falha <- falso

repetir

C <- condição de SC

Provar (C)

se não foi possível provar C **então**

Falha <- verdadeiro

até não existir mais nenhuma condição em SC **ou** Falha == verdadeiro

até não existir mais nenhuma regra em SR **ou** Falha == falso

se Falha == falso **então**

Q é verdadeiro e provar (Q) teve sucesso

senão

Q é falso e provar (Q) falhou

Encadeamento para Trás

- Características
 - Raciocínio orientado por metas;
 - Útil para responder perguntas como: “O que fazer agora?”
 - Frequentemente custo muito menor que um custo linear em relação ao tamanho da base de conhecimento;

Dúvidas?