# Agentes Lógicos

Função Agente
Definições de Inferência
Algoritmo de Enumeração da Tabela Verdade (TV)
Equivalencia, Validade, Satisfatibilidade, Monotonicidade
Redução ao absurdo
Resolução e Cláusula
Forma Normal Conjuntiva
Completeza
Inferencia Unica ou Resolução
Algoritmo de Resolução

# Função Agente

```
funcao AgenteBC(percepcao) : acao
    BC: BaseDeConhecimento
    t    : contador;
    // Informe o que percebe
    TELL(BC, CriarSentencaDePercepcao(percepcao,t))
    //pergunte que acao fazer
   acao<-- ASK(BC, CriarConsultaAcao(t));
   //Informe o que foi executado
  TELL(BC,CriarSentencaDeAcao(acao,t))
   t++;
```

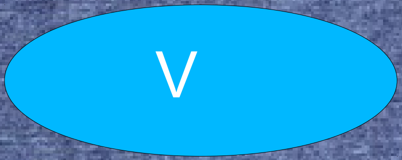
# Definições de Inferência

- Derivação de novas sentenças a partir de sentenças antigas
- Passagem de uma proposição à outra
- Passagem através de regras válidas do antecedente ao consequente

# Equivalência, validade, satisfatibilidade, monotonicidade

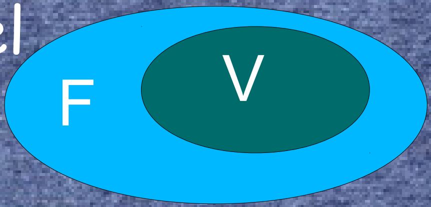
- Equivalencia logica: Duas sentenças A e B são logicamente equivalentes se são verdadeiras no mesmo conjunto de modelos.

- Validade: Uma sentença é válida se é verdadeira em todos os modelos (tautologia)




- Satisfatibilidade: A sentença é verdadeira em algum modelo. A é valida sss $\neg A$ é não satisfativel

  A é satisfativel sss $\neg A$ é não valida




- Monotonicidade: Informações adicionais não invalidam o que já se sabe. Se $BC \models A$, então $BC \land B \models A$

# Redução ao absurdo

- Também chamada de prova por refutação ou prova por contradição ("reductio ad absurdum")

  $\alpha \models \beta$ sss $(\alpha \wedge \neg\beta)$ é não satisfatível (falso).

- Nega-se $\beta$ e mantém-se $\alpha$ chegando a uma contradição.
- De outro modo, $\alpha \rightarrow F$, só é verdadeiro se $\alpha$ for F.
- Verdade não implica em falsidade

# Resolução unitária, ou regra de inferência única

- Tem a ver com a completeza (garantia de encontrar uma resolução quando esta existir)
- CLÁUSULA: É uma disjunção de literais
- Esta regra pega uma cláusula e um literal m e produz uma nova cláusula. Esse m é negação de algum $l_i$

$l_1 \vee l_2 \vee \ldots l_k, \quad m = \neg l_i$

------------------------------,     perceba que $l_i$ sumiu embaixo

$l_1 \vee l_2 \vee \ldots \; l_{i-1} \vee l_{i+1} \vee \ldots \quad l_k$

# Resolução geral

- Esta regra pega uma cláusula e um literal m e produz uma nova cláusula. Esse $m_j$ é negação de algum $l_i$

$$l_1 \lor l_2 \lor \ldots\ l_k, \quad m_1 \lor m_2 \lor \ldots m_n$$

------------------------------------------------------------------------

$$l_1 \lor l_2 \lor \ldots l_{i-1} \lor l_{i+1} \lor ..\ l_k \ \lor \ m_1 \lor m_2 \lor \ldots \lor \ldots m_{j-1} \lor m_{j+1} \lor \ldots m_n$$

- Perceba que $m_j$ e $l_i$ sumiram embaixo.

# Forma Normal Conjuntiva

- É uma conjunção de disjunções
- Toda sentença lógica proposicional é equivalente a uma conjunção de disjunções de literais
- Aplicando regras usuais (De Morgan, Eliminação de ==>, etc) podemos transformar qualquer sentença em FNC
- Podemos aplicar a regra de resolução dentro das disjun-ções de literais.
- $(l_{1,1} \vee .. \vee l_{1,k}) \wedge ... (l_{n,1} \vee .. \vee l_{n,k})$

# Algoritmo de Resolução da Lógica Proposicional (LP)

Este algoritmo usa o princípio da contradição.

Para mostrar que BC |= $\alpha$ mostramos que (BC$\wedge\neg\alpha$) é não satisfatível (satisfativel = verdadeiro em alguns modelos)

- 1o) BC$\wedge\neg\alpha$ é convertido em FNC.
- 2o) Aplica-se a regra de resolução às clausulas resultantes. Cada par que contém literais complementares (negação de outros literais) é resolvido para gerar uma nova cláusula, que é adicionada ao conjunto se ainda não estiver presente.

# Algoritmo de Resolução da Lógica Proposicional (LP)

- O processo continua até acontecer dois fatos:

1)Não existir nenhuma nova cláusula a ser adicionada. A não tem B como consequência logica

2)Uma aplicação de resolução derivar em cláusula vazia; nesse caso A tem como consequência logica B.

**Cláusula vazia – uma disjunção de nenhum disjunto é equivalente a FALSO. Uma disjunção só é verdadeiro se pelo menos um de seus disjuntos é verdadeiro.**

**Outro modo de ver que uma cláusula vazia representa uma contradição é que ela só surge da solução de duas cláusulas unitárias complementares $P \wedge \neg P$**

# Algoritmo de Resolução da Lógica Proposicional (LP)

funcao ResolucaoLP(BC,c)

c: consulta, em logica proposicional

clausulas<-- conjunto de todas as clausulas em FNC $\wedge \neg$c

nova <--{ }

repita

para Ci e Cj em clausulas faça

resolventes ← ResolverLP(Ci,Cj)

se resolventes contem a clausula vazia retorne V

nova ← nova U resolventes

se nova$\subseteq$ clausulas retorne F

clausulas ← clausulas U nova

**Cláusula vazia – uma disjunção de nenhum disjunto é equivalente a FALSO. Uma disjunção só é verdadeiro se pelo menos um de seus disjuntos é verdadeiro.**

**Outro modo de ver que uma cláusula vazia representa uma contradição é que ela só surge da solução de duas cláusulas unitárias complementares P∧¬P**

# Aplicação ResoluçãoLP

- Dados:

$BC = R_2 \wedge R_4$

**Consulta** : $\neg P_{1,2}$ ?

$R_2$: $B_{1,1} \Leftrightarrow (P_{1,2} \wedge P_{2,1})$

$R_4$: $\neg B_{1,1}$

Resolver $BC \wedge \neg(\neg P_{1,2})$ ou seja, $BC \wedge P_{1,2}$

# Converter $R_2$ em FNC

$R_2$: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

- Eliminar $\Leftrightarrow$, trocando $\alpha \Leftrightarrow \beta$ por $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
  $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminar $\Rightarrow$, trocando $\alpha \Rightarrow \beta$ por $\neg \alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Mover $\neg$ para dentro usando as leis de de Morgan e negação dupla:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \vee \neg P_{2,1}) \vee B_{1,1})$

4. Aplicar a lei distributiva ($\wedge$ sobre $\vee$) e eliminar '(' ')':
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

# Aplicação ResoluçãoLP(BC,c)

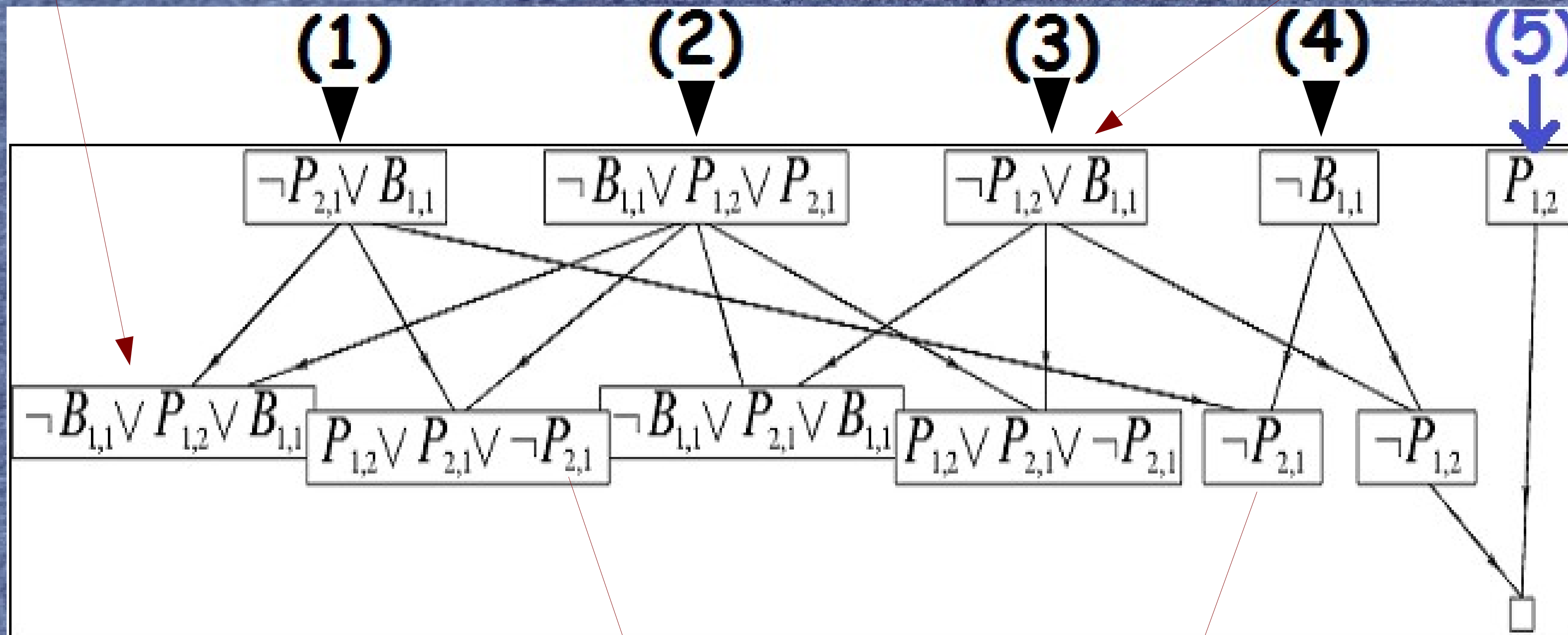
APLICAR ResolucaoLP(BC,c) PARA AS CLÁUSULAS

$$\underbrace{[(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})]}_{BC} \wedge \underbrace{P_{1,2}}_{\neg C}$$

# Aplicação ResoluçãoLP(BC,c)

resolventes

cláusulas




**Erro livro: estava como $\neg P_{12}$**

**Erro livro: resolvente $\neg P_{21}$ era de (1), (2), (4), ao invés de (1), (4)**

# Aplicação ResoluçãoLP(BC,c)