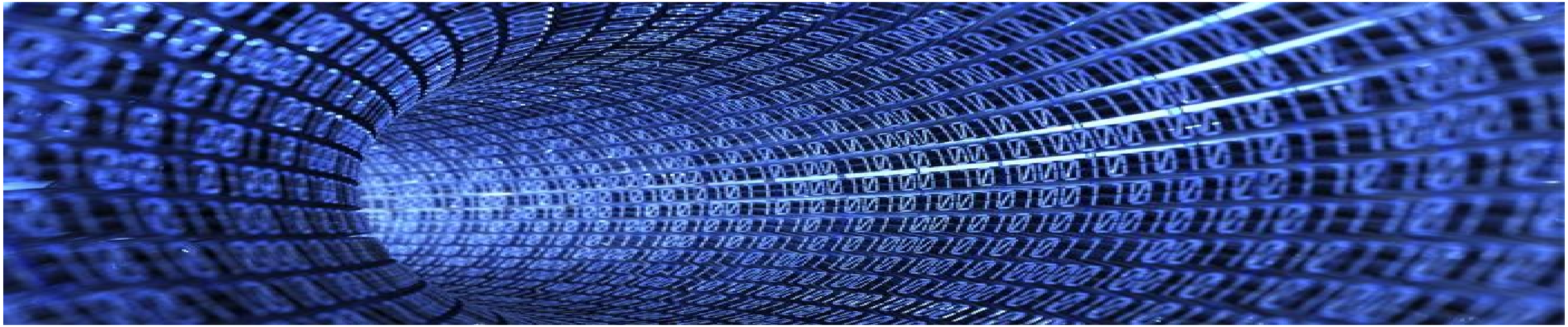
INSTITUTO MAUÁ DE TECNOLOGIA

Curso de Engenharia de Computação
Linguagens Formais, Autômatos e Compiladores

Lógica Proposicional



Slides da disciplina ECM253 – Linguagens Formais, Autômatos e Compiladores
Curso de Engenharia de Computação
Instituto Mauá de Tecnologia – Escola de Engenharia Mauá
Prof. Marco Antonio Furlan de Souza

Lógica proposicional

- **Conceitos**
  - **Utiliza** a **lógica formal** para **descobrir** como **obter conclusões lógicas** a **partir** de **sentenças** existentes;
  - As **sentenças** também são denominadas de **proposições**;
  - **Lógica proposicional** também é conhecida por **lógica de sentenças** ou **cálculo proposicional**.

Lógica proposicional

- **Argumentos válidos**
  - **Representação**:

$$\boxed{P_1 \wedge P_2 \wedge P_3 \wedge \ldots \wedge P_n \rightarrow Q}$$

  - P_1, P_2, … ,P_n são **sentenças** denominadas de **hipóteses** do **argumento** e Q é a **conclusão**. Os P_i e Q são **fbfs**. **Lê-se**:
    - "Q é uma **conclusão lógica** de P_1, P_2, … ,P_n sempre que a verdade de P_1, P_2, … ,P_n implica na verdade de Q."
  - A fbf P_1, P_2, … ,$P_n \rightarrow Q$ é um **argumento válido** quando é uma **tautologia**.
  - Para determinar uma tautologia:
    - Criar uma **tabela verdade**;
    - Utilizar um **algoritmo** de **teste** de **tautologia**.

Lógica proposicional

- **Argumentos válidos**
  - Algoritmo para testar tautologia

```
TestarTautologia(fbf P; fbf Q)
//Dados fbfs P e Q, decidir se a fbf P→Q é uma tautologia
início
    //Assumir que P→Q NÃO é uma tautologia
    P = true //atribuir V para P
    Q = false    //atribuir F para Q
    repita
        Para cada fbf composta que já tenha um valor verdade
        atribuído, atribua valores verdade a seus componentes
    até que todas as ocorrências de símbolos tenham valores
         verdade atribuídos
    se algum símbolo possui dois valores verdade
    então //Há uma contradição – é falso que não é tautologia
        escreva('P→Q É uma tautologia')
    senão //Provou-se que é verdade que não é uma tautologia
        escreva('P→Q NÃO É uma tautologia')
    fim se
fim TestarTautologia
```

Lógica proposicional

- **Argumentos válidos**
  - **Exemplo**
    - “Se George Washington foi o primeiro presidente dos Estados Unidos, então John Adams foi o primeiro vice-presidente. George Washington foi o primeiro presidente dos Estados Unidos. Portanto, John Adams foi o primeiro vice-presidente.”
    - Este argumento possui **duas hipóteses**:
      - “Se George Washington foi o primeiro presidente dos Estados Unidos, então John Adams foi o primeiro vice-presidente.”($A \to B$)
      - “George Washington foi o primeiro presidente dos Estados Unidos.”(A)
    - E a **conclusão**:
      - “John Adams foi o primeiro vice-presidente.”(B)

Lógica proposicional

- **Argumentos válidos**
  - **Exemplo**
    - **Representação simbólica** do argumento:

$$\boxed{(A \rightarrow B) \wedge A \rightarrow B}$$

    - **Prova** com o algoritmo `TestarTautologia`:
      - Atribuir `V` à $A \rightarrow B \wedge A$;
      - Atribuir `F` à B;
      - Primeira repetição:
        - $A \rightarrow B$ é `V` e A é `V` (consequência da conjunção)
      - Segunda repetição:
        - Se $A \rightarrow B$ é `V` então é o caso que A deva ser `V` ou `F` e B não pode ser `F`, então atribui-se `V` à B;
        - Mas A é `V` e B é `F` (início e primeira repetição) – então B é valorado com `V` e `F`.
      - Como B ficou com **dois valores** – a prova que não é tautologia falha – TEMOS UMA TAUTOLOGIA!

Lógica proposicional

- **Argumentos válidos**
  - **Exemplo**
    - **Representação simbólica** do argumento:

$$\boxed{(A \rightarrow B) \wedge A \rightarrow B}$$

    - **Prova** com tabela verdade:

A	B	$A \rightarrow B$	$(A \rightarrow B) \wedge A$	$(A \rightarrow B) \wedge A \rightarrow B$
V	V	V	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	V

Tautologia

Lógica proposicional

- **Sequência de prova**
  - É uma **sequência** de **fbfs** na qual **cada fbf ou é** uma **hipótese ou é** o **resultado** da **aplicação de uma** das **regras** de **derivação** de um **sistema formal** à **fbfs existentes** na sequência;
  - Esta técnica permite a **verificação** de **tautologias**:

$$
\begin{array}{rl}
P_1 & \text{(hipótese)} \\
P_2 & \text{(hipótese)} \\
\vdots & \\
P_n & \text{(hipótese)} \\
\text{fbf}_1 & \text{(obtida pela aplicação de regra de derivação)} \\
\text{fbf}_2 & \text{(obtida pela aplicação de regra de derivação)} \\
\vdots & \\
Q & \text{(obtida pela aplicação de regra de derivação)}
\end{array}
$$

Lógica proposicional

- **Regras de derivação**
  - Devem ser **escolhidas** de modo que o **sistema formal** seja **correto** (somente argumentos válidos podem ser provados) e **completo** (todo argumento válido deveria ser provado);
  - Duas **categorias** para **regras** de **derivação**:
    - **Regras de equivalência**: permitem que fbfs individuais sejam reescritas;
    - **Regras de inferência**: permitem que novas fbfs sejam derivadas a partir de fbfs já existentes na sequência de prova.

Lógica proposicional

- **Regras de derivação**
  - **Regras de equivalência**
    - Estabelece que certos **pares** de fbfs são **equivalentes**.

Regras de equivalência		
Expressão	**Equivalente à**	**Nome/abreviação**
$P \lor Q$ $P \land Q$	$Q \lor P$ $Q \land P$	Comutativa/com
$(P \lor Q) \lor R$ $(P \land Q) \land R$	$P \lor (Q \lor R)$ $P \land (Q \land R)$	Associativa/ass
$\neg(P \lor Q)$ $\neg(P \land Q)$	$\neg P \land \neg Q$ $\neg P \lor \neg Q$	Leis de DeMorgan/ DeMorgan
$P \rightarrow Q$	$\neg P \lor Q$	Implicação/imp
P	$\neg(\neg P)$	Negação dupla/dn
$P \leftrightarrow Q$	$(P \rightarrow Q) \land (Q \rightarrow P)$	Definição de equivalência/equ

Lógica proposicional

- **Regras de derivação**
  - **Regras de equivalência**
    - **Exemplo** de aplicação

$$
\begin{array}{lr}
1.\ (\neg A \vee \neg B) \vee C & \text{(hipótese)} \\
2.\ \neg(A \wedge B) \vee C & \text{1, DeMorgan} \\
3.\ (A \wedge B) \rightarrow C & \text{2, imp}
\end{array}
$$

Lógica proposicional

- **Regras de derivação**
  - **Regras de inferência**
    - Estabelece que **se uma** ou **mais fbfs** existentes **correspondem** com a **primeira parte** de um **padrão** de **regra**, **então** pode-se **adicionar** à **sequência** uma **nova fbf produzida** pela **segunda parte** do **padrão** que foi correspondido.

Modus Ponens – "Método de AFIRMAR"

Regras de inferência		
De	**Pode derivar**	**Nome/abreviação**
P, P→Q	Q	*Modus ponens*/mp
P→Q, ¬Q	¬P	*Modus tollens*/mt
P, Q	P∧Q	Conjunção/con
P∧Q	P, Q	Simplificação/sim
P	P∨Q	Adição/add

Modus tollens – "Método DE NEGAR"

Lógica proposicional

- **Regras de derivação**
  - **Regras de inferência**
    - **Exemplo** de aplicação

$1.\ A \rightarrow (B \wedge C)$	(hipótese)
$2.\ A$	(hipótese)
$3.\ B \wedge C$	1,2, mp

Lógica proposicional

- **Teste seus conhecimentos**
  1) Usando a lógica proposicional, demonstre o teorema:

$$\boxed{\neg P \to (P \to Q)}$$

  2) Traduzir em símbolos e provar: “Se o programa é eficiente, ele executará rapidamente. Ou o programa é eficiente ou ele tem um erro. No entanto, o programa não executa rapidamente. Portanto o programa tem um erro. (Utilizar os símbolos E (eficiente), Q (rápido), B (erro)).
  3) Verificar com tabela-verdade que:

$$\boxed{A \to (B \to C) \Leftrightarrow (A \wedge B) \to C}$$

     É uma tautologia.