Introdução à Lógica Computacional

Circuitos: Maps de Karnaugh

Lógica Proposicional: Prova por Refutação

Agenda da aula

- Circuitos lógicos: Mapas de Karnaugh
- Recaptulando semântica da lógica proposicional
  - Representação
  - Prova por tabela verdade
  - Prova por dedução
- Prova por refutação
- Próxima aula
  - Exercícios
  - Tableau

Simplificação de circuitos lógicos

1. $ABC + \overline{A}B\overline{C}$
2. $AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$
3. $\overline{A}B + C\overline{D} + EF + GK + H\overline{L}$

- Representação grammatical
  - A técnica de simplificação que será utilizada requer que a expressão esteja na forma de soma de produtos
  - Forma de soma de produtos:
  - Uma barra não pode cobrir mais que uma variável em um termo

A	B	X
0	0	1 → $\bar{A}\bar{B}$
0	1	0
1	0	0
1	1	1 → AB

$$\left\{ x = \bar{A}\bar{B} + AB \right\}$$

	$\bar{B}$	B
$\bar{A}$	1	0
A	0	1

Mapa de Karnaugh

- Método gráfico usado para simplificar uma equação lógica ou converter uma tabela verdade no seu circuito lógico correspondente

A	B	C	X	
0	0	0	1	$\rightarrow \bar{A}\bar{B}\bar{C}$
0	0	1	1	$\rightarrow \bar{A}\bar{B}C$
0	1	0	1	$\rightarrow \bar{A}B\bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$\rightarrow AB\bar{C}$
1	1	1	0	

$$\left\{ \begin{aligned} X = {} & \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C \\ & + \bar{A}B\bar{C} + AB\bar{C} \end{aligned} \right\}$$

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	0
AB	1	0
$A\bar{B}$	0	0

Mapa de Karnaugh com 3 variáveis

- Para 3 e 4 variáveis usar o código Gray: de um número para outro, apenas um bit varia

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\rightarrow \bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	

$$\left\{ \begin{aligned} X = {} & \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D \\ & + AB\bar{C}D + ABCD \end{aligned} \right\}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

Mapa de Karnaugh com 4variáveis

- Agrupamentos possíveis

Mapa de Karnaugh: método

Passo 1 Construa o mapa K e coloque os 1s nos quadros que correspondem aos 1s na tabela-verdade. Coloque 0s nos outros quadros.

Passo 2 Analise o mapa quanto aos 1s adjacentes e agrupe os 1s que *não* sejam adjacentes a quaisquer outros 1s. Esses são denominados 1s *isolados*.

Passo 3 Em seguida, procure os 1s que são adjacentes a somente um outro 1. Agrupe *todo* par que contém tal 1.

Passo 4 Agrupe qualquer octeto, mesmo que ele contenha alguns 1s que já tenham sido agrupados.

Passo 5 Agrupe qualquer quarteto que contenha um ou mais 1s que ainda não tenham sido agrupados, *certificando-se de usar o menor número de agrupamentos*.

Passo 6 Agrupe quaisquer pares necessários para incluir quaisquer 1s que ainda não tenham sido agrupados, *certificando-se de usar o menor número de agrupamentos*.

Passo 7 Forme a soma OR de todos os termos gerados por cada grupo.

Exemplos

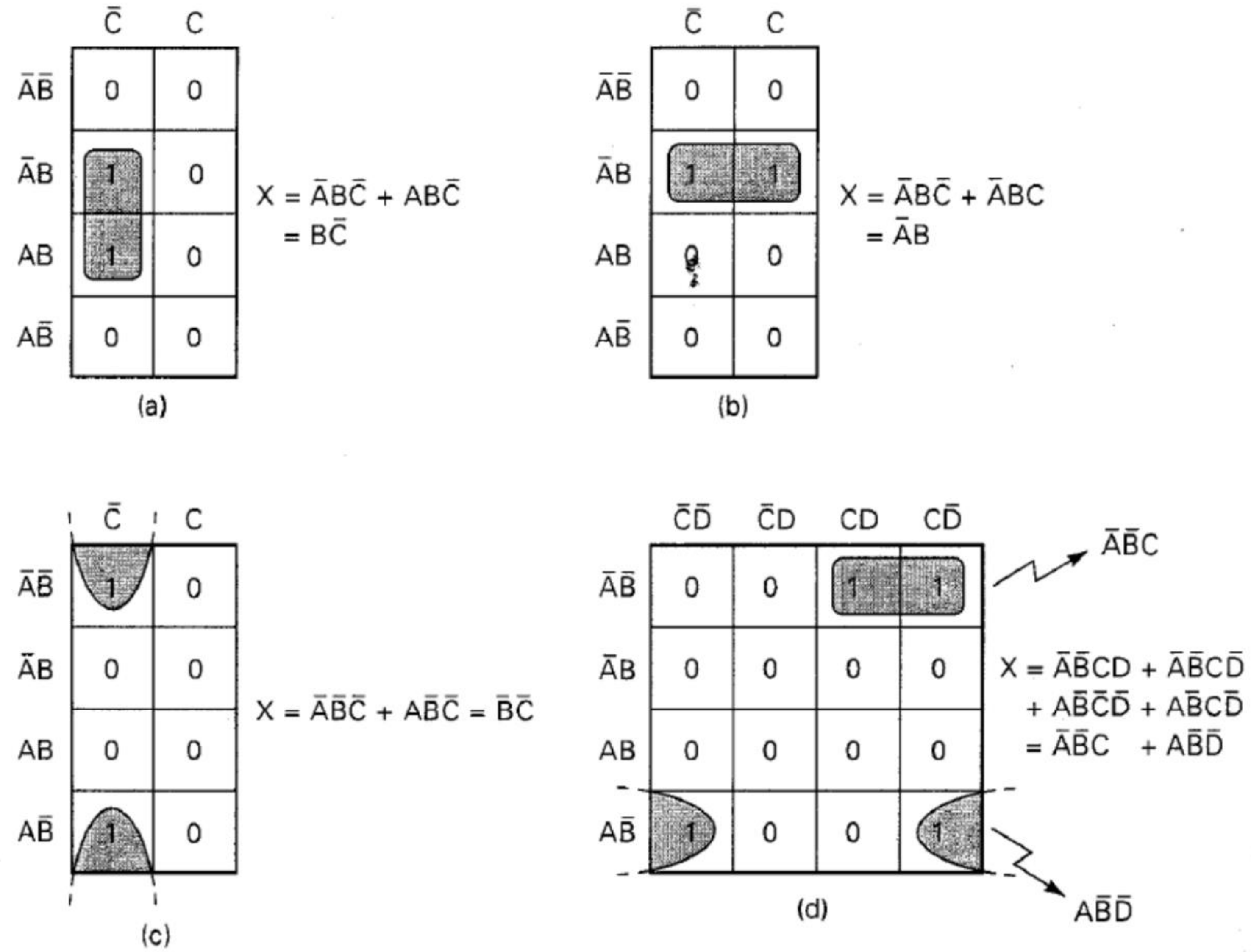
$$X = \overline{A}B\overline{C} + AB\overline{C}$$

$$X = \overline{A}B\overline{C} + \overline{A}BC$$

$$X = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C}$$

$$X = \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D}$$

Solução



$X = \bar{A}B\bar{C} + AB\bar{C}$
$= B\bar{C}$
(a)
$X = \bar{A}B\bar{C} + \bar{A}BC$
$= \bar{A}B$
(b)
$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{B}\bar{C}$
(c)
$\bar{A}\bar{B}C$
$X = \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D}$
$+ A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$
$= \bar{A}\bar{B}C \quad + A\bar{B}\bar{D}$
$A\bar{B}\bar{D}$
(d)

Representação em lógica proposcisional: Recordar é viver....

- Usando a sintaxe da lógica proposicional, formalize o argumento:
  - Se o time joga bem, então ganha o campeonato.
  - Se o time não joga bem, então o técnico é culpado.
  - Se o time ganha o campeonato, então os torcedores ficam contentes.
  - Os torcedores não estão contentes. Logo, o técnico é culpado.

Usando sua intuição

Qual(is) dos argumentos a seguir é válido?

Argumento 1

- Se eu fosse artista, então eu seria famoso.
- Não sou famoso.
- Logo, não sou artista.

Argumento 2

- Se eu fosse artista, então eu seria famoso.
- Sou famoso.
- Logo, sou artista

Mostrar a validade do argumento abaixo

- Exemplo: validar o argumento {j→g, ¬j→t, g→c, ¬c} |-- t

Por tabela verdade

j	g	t	c	~j	~c	j-->g	~j--->t	g--> c	(j-->g) E (~j-->t) E (g-->c) E (~c)	((j-->g) E (~j-->t) E (g-->c) E (~c)) --> t
V	V	V	V	F	F	V	V	V	F	V
V	V	V	F	F	V	V	V	F	F	V
V	V	F	V	F	F	V	V	V	F	V
V	V	F	F	F	V	V	V	F	F	V
V	F	V	V	F	F	F	V	V	F	V
V	F	V	F	F	V	F	V	V	F	V
V	F	F	V	F	F	F	V	V	F	V
V	F	F	F	F	V	F	V	V	F	V
F	V	V	V	V	F	V	V	V	F	V
F	V	V	F	V	V	V	V	F	F	V
F	V	F	V	V	F	V	F	V	F	V
F	V	F	F	V	V	V	F	F	F	V
F	F	V	V	V	F	V	V	V	F	V
F	F	V	F	V	V	V	V	V	F	V
F	F	F	V	V	F	V	F	V	F	V
F	F	F	F	V	V	V	F	V	F	V

Mostrar a validade do argumento abaixo

- Exemplo: validar o argumento {j→g, ¬j→t, g→c, ¬c} |-- t

Por dedução

1. j→g
2. ~j → t
3. g→c
4. ˜c

------

5. t

6) j→c (1) E (3) silogismo hipotético

7) ~j (6) E (4) MODUS TOLLENS

8) t (7) e (2) MODUS PONENS

PROVA POR REFUTAÇÃO

- CONSIDERE O ARGUMENTO ABAIXO

(1) Se o time joga bem, então ganha o campeonato.

(2) Se o time não joga bem, então o técnico é culpado.

(3) Se o time ganha o campeonato, então os torcedores ficam contentes.

(4) Os torcedores não estão contentes.

(5) Logo, o técnico é culpado.

PROVA POR REFUTAÇÃO

- CONSIDERE O ARGUMENTO ABAIXO

(1) Se o time joga bem, então ganha o campeonato.

(2) Se o time não joga bem, então o técnico é culpado.

(3) Se o time ganha o campeonato, então os torcedores ficam contentes.

(4) Os torcedores não estão contentes.

(5) Logo, o técnico é culpado.

Refutação

(a) O técnico **não** é culpado **hipótese**

PROVA POR REFUTAÇÃO

- CONSIDERE O ARGUMENTO ABAIXO

(1) Se o time joga bem, então ganha o campeonato.

(2) Se o time não joga bem, então o técnico é culpado.

(3) Se o time ganha o campeonato, então os torcedores ficam contentes.

(4) Os torcedores não estão contentes.

(5) Logo, o técnico é culpado.

Refutação

(a) O técnico **não** é culpado	**hipótese**
(b) O time joga bem	MODUS TOLLENS (a,2)

PROVA POR REFUTAÇÃO

- CONSIDERE O ARGUMENTO ABAIXO

(1) Se o time joga bem, então ganha o campeonato.

(2) Se o time não joga bem, então o técnico é culpado.

(3) Se o time ganha o campeonato, então os torcedores ficam contentes.

(4) Os torcedores não estão contentes.

(5) Logo, o técnico é culpado.

Refutação

(a) O técnico **não** é culpado	**hipótese**
(b) O time joga bem	MODUS TOLLENS (a,2)
(c) O time ganha o campeonato	MODUS PONENS (b,1)

PROVA POR REFUTAÇÃO

- CONSIDERE O ARGUMENTO ABAIXO

(1) Se o time joga bem, então ganha o campeonato.

(2) Se o time não joga bem, então o técnico é culpado.

(3) Se o time ganha o campeonato, então os torcedores ficam contentes.

(4) Os torcedores não estão contentes.

(5) Logo, o técnico é culpado.

Refutação

(a) O técnico **não** é culpado	**hipótese**
(b) O time joga bem	MODUS TOLLENS (a,2)
(c) O time ganha o campeonato	MODUS PONENS (b,1)
(d) Os torcedores ficam contentes	MODUS PONENS (c,3)

PROVA POR REFUTAÇÃO

- CONSIDERE O ARGUMENTO ABAIXO

(1) Se o time joga bem, então ganha o campeonato.

(2) Se o time não joga bem, então o técnico é culpado.

(3) Se o time ganha o campeonato, então os torcedores ficam contentes.

(4) Os torcedores não estão contentes.

(5) Logo, o técnico é culpado.

Refutação

(a) O técnico **não** é culpado	**hipótese**
(b) O time joga bem	MODUS TOLLENS (a,2)
(c) O time ganha o campeonato	MODUS PONENS (b,1)
(d) Os torcedores ficam contentes	MODUS PONENS (c,3)
(e) Contradição!	Confrontando (d) e (4)

Conclusão: a hipótese contradiz as premissas, logo o argumento é válido!

PROVA POR REFUTAÇÃO

- Voltando ao nosso exemplo: validar o argumento {j→g, ¬j→t, g→c, ¬c} |--t
- (1) j→g
- (2) ¬j→t
- (3) g→c
- (4) ¬c
- ------------
- (5) t

PROVA POR REFUTAÇÃO

- (5)¬t Hipótese
- (6) j MT(5,2)
- (7) g MP(6,1)
- (8) c MP(7,3)
- (9). $\square$ Contradição!
- Conclusão: como {premissas} U {¬t} é inconsistente, segue que {premissas} |-- t

FORMA NORMAL CONJUNTIVA

- Para simplificar a automatização do processo de refutação, usa-se fórmulas normais (Forma Normal Conjuntiva - FNC)
  - Todas as sentenças são descritas como uma conjunção de disjunções, isto é, cada sentença só pode ter conectivo OU ou a negação de proposições primitivas
  - Elimine a implicação:
    - $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$
  - Reduza o escopo da negação:
    - $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$
    - $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$
  - Reduza o escopo da disjunção:
    - $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Exemplo de FNC

- **$p \vee q \rightarrow r \wedge s$**

Exemplo de FNC

- **$p \lor q \rightarrow r \land s$**

$\equiv \neg (p \lor q) \lor (r \land s)$

Exemplo de FNC

- **$p \lor q \rightarrow r \land s$**

$\equiv \neg (p \lor q) \lor (r \land s)$

$\equiv (\neg p \land \neg q) \lor (r \land s)$

Exemplo de FNC

- **$p \lor q \rightarrow r \land s$**

$\equiv \neg (p \lor q) \lor (r \land s)$

$\equiv (\neg p \land \neg q) \lor (r \land s)$

$\equiv ((\neg p \land \neg q) \lor r) \land ((\neg p \land \neg q) \lor s)$

Exemplo de FNC

- **p∨ q → r ∧ s**

$\equiv \neg (p \vee q) \vee (r \wedge s)$

$\equiv (\neg p \wedge \neg q) \vee (r \wedge s)$

$\equiv ((\neg p \wedge \neg q) \vee r) \wedge ((\neg p \wedge \neg q) \vee s)$

$\equiv (\neg p \vee r) \wedge (\neg q \vee r) \wedge (\neg p \vee s) \wedge (\neg q \vee s)$

Fórmulas normais:

$\{\neg p \vee r, \neg q \vee r, \neg p \vee s, \neg q \vee s\}$

Inferência por Resolução

- FNC permite usar inferência por resolução
- A idéia da resolução é:

$$\text{RES}(\alpha \vee \beta, \neg \beta \vee \gamma) = \alpha \vee \gamma$$

$$\text{RES}(\alpha, \neg \alpha) = \square$$

Inferência por Resolução

- FNC permite usar inferência por resolução
- A idéia da resolução é:

$$RES(\alpha \lor \beta, \neg \beta \lor \gamma) = \alpha \lor \gamma$$

$$RES(\alpha, \neg \alpha) = \square$$

Equivalências:

$MP(\alpha \to \beta, \alpha) = \beta$	$RES(\neg\alpha \lor \beta, \alpha) = \beta$
$MT(\alpha \to \beta, \neg\beta) = \neg\alpha$	$RES(\neg\alpha \lor \beta, \neg\beta) = \neg\alpha$
$SH(\alpha \to \beta, \beta \to \gamma) = \alpha \to \gamma$	$RES(\neg\alpha \lor \beta, \neg\beta \lor \gamma) = \neg\alpha \lor \gamma$

Voltando ao exemplo

- {j→g, ¬j→t, g→c, ¬c} |-- t
- Passo 1: Colocar na FNC

(1) ¬j∨g

(2) j∨t

(3) ¬g∨c

(4) ¬c

- Passo 2: Negar a conclusão (Hipótese)

(5) ¬t Hipótese

- Passo 3: Aplicar as regras de inferência

(6) j RES(5,2)

(7) g RES(6,1)

(8) c RES(7,3)

(9) Absurdo □ RES(8,4)

Conclusão: como {premissas} ∪ {¬t} é inconsistente, segue que {premissas}m |-- t .

Exercício

- Prove por refutação a validade do argumento abaixo

Se o programa possui erros de sintaxe, sua compilação produz mensagem de erro.

Se o programa não possui erros de sintaxe, sua compilação produz um executável.

Se tivermos um programa executável, podemos executá-lo para obter um resultado.

Não temos como executar o programa para obter um resultado.

Logo, a compilação do programa produz uma mensagem de erro.

Exercício

- Prove por refutação a validade do argumento abaixo

Sentenças primitivas

P1= o programa possui erros de sintaxe

P2= a compilação do programa produz mensagem de erro.

P3= a compilação do programa produz um executável.

P4 = podemos executar o programa para obter um resultado.

Exercício

- Prove por refutação a validade do argumento abaixo

Sentenças primitivas

P1= o programa possui erros de sintaxe

P2= a compilação do programa produz mensagem de erro.

P3= a compilação do programa produz um executável.

P4 = podemos executar o programa para obter um resultado.

Representação em lógica proposicional

1. $P1 \rightarrow P2$
2. $\sim P1 \rightarrow P3$
3. $P3 \rightarrow P4$
4. $\sim P4$

----

5. P2

Exercício

- Prove por refutação a validade do argumento abaixo

Sentenças primitivas

P1= o programa possui erros de sintaxe

P2= a compilação do programa produz mensagem de erro.

P3= a compilação do programa produz um executável.

P4 = podemos executar o programa para obter um resultado.

Representação

1. $P1 \rightarrow P2$
2. $\sim P1 \rightarrow P3$
3. $P3 \rightarrow P4$
4. $\sim P4$

----

5. P2

Representação em FNC

1. $\sim P1 \vee P2$
2. $P1 \vee P3$
3. $\sim P3 \vee P4$
4. $\sim P4$

Exercício

- Prove por refutação a validade do argumento abaixo

Sentenças primitivas

P1= o programa possui erros de sintaxe

P2= a compilação do programa produz mensagem de erro.

P3= a compilação do programa produz um executável.

P4 = podemos executar o programa para obter um resultado.

Representação

1. $P1 \rightarrow P2$
2. $\sim P1 \rightarrow P3$
3. $P3 \rightarrow P4$
4. $\sim P4$

----

5. P2

FNC

1. $\sim P1 \vee P2$
2. $P1 \vee P3$
3. $\sim P3 \vee P4$
4. $\sim P4$

Negando a conclusão

5. $\sim P2$

Exercício

- Prove por refutação a validade do argumento abaixo

Sentenças primitivas

P1= o programa possui erros de sintaxe

P2= a compilação do programa produz mensagem de erro.

P3= a compilação do programa produz um executável.

P4 = podemos executar o programa para obter um resultado.

Representação

1. P1 → P2
2. ~P1 → P3
3. P3 → P4
4. ~ P4

----

5. P2

FNC

1. ~P1 v P2
2. P1 v P3
3. ~ P3 v P4
4. ~ P4

Negando a conclusão

5. ~P2

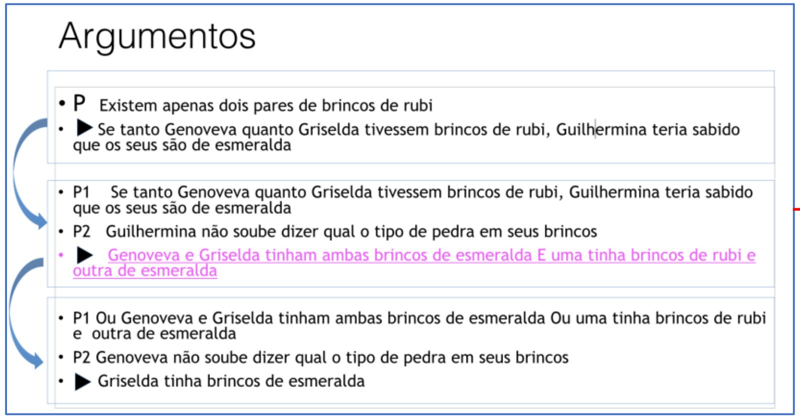
Aplicando as regras de resolução

6.	~P1.	RES (1) e (5)
7.	P3	RES (6) e (2)
8.	P4	RES (7) e (3)
9.	ABSURDO	(8) E (4)

Tarefa: Ponto extra na média, caso seja necessário para fazer prova final

- Objetivo:
  - Fomentar uma revisão da matéria e estudar o material apresentado em aula
  - Fomentar a participação em sala de aula
- Tarefa:
  - Toda aula haverá um slide com um pequeno erro de lógica. Tal erro será falado em sala de aula.
  - Apresentar o slide com erro e o consertado
  - É necessário que se apresente 1 slide por aula para que o aluno ganhar o ponto extra
- Obs.: Só valerá o ponto extra na média, se o aluno apresentar pelo menos um slide por aula, mas de TODAS as aulas.
- Exemplo de entrega de 1 aula

Aula 1: erro marcado em rosa



Aula 1: slide corrigido em vermelho