Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

ACH2013 – Matemática Discreta – 2º sem. 2021
Professor: José Ricardo G. Mendonça

5ª Lista de Exercícios – Álgebras booleanas – 17 nov. 2021

In the choice of component ideas, as in the choice of expressions, the aim must be to convey the greatest quantity of thoughts with the smallest quantity of words.
Herbert Spencer, *Philosophy of Style: An Essay* (1884)

I. Álgebras booleanas

1. Escreva a forma dual das seguintes expressões booleanas:

   (*a*) $ab + c'$;

   (*b*) $bc + ca$;

   (*c*) $a'bc$;

   (*d*) $a(b + c')b$;

   (*e*) $ab + bc + ca$;

2. Seja $\mathscr{B}$ uma álgebra definida pelo conjunto $B = \{0, a, b, c, d, e, f, 1\}$ e pelas seguintes operações sobre os elementos de B:

$+$	0	a	b	c	d	e	f	1
0	0	a	b	c	d	e	f	1
a	a	a	c	c	e	e	1	1
b	b	c	b	c	f	1	f	1
c	c	c	c	c	1	1	1	1
d	d	e	f	1	d	e	f	1
e	e	e	1	1	e	e	1	1
f	f	1	f	1	f	1	f	1
1	1	1	1	1	1	1	1	1

$*$	0	a	b	c	d	e	f	1
0	0	0	0	0	0	0	0	0
a	0	a	0	a	0	a	0	a
b	0	0	b	b	0	0	b	b
c	0	a	b	c	0	a	b	c
d	0	0	0	0	d	d	d	d
e	0	a	0	a	d	e	d	e
f	0	0	b	b	d	d	f	f
1	0	a	b	c	d	e	f	1

x	x'
0	1
a	f
b	e
c	d
d	c
e	b
f	a
1	0

   Mostre que $\mathscr{B} = \langle B, +, *, ', 0, 1\rangle$ é uma álgebra booleana.

3. Seja $D_N \subset \mathbb{N}$ o conjunto dos divisores positivos do número N e sejam as operações sobre os elementos de D_N definidas por $a + b = \text{mmc}(a, b)$, $a * b = \text{mdc}(a, b)$ e $a' = N/a$.

(*a*) Mostre que para as operações '$+$' e '$*$' definidas acima o elemento nulo aditivo é dado por $1 \in D_N$ e o elemento nulo multiplicativo é dado por $N \in D_N$;

(*b*) Mostre que $\mathscr{B} = \langle D_{15}, +, *, ', 1, 15\rangle$ é uma álgebra booleana;

(*c*) Mostre que $\mathscr{B} = \langle D_{70}, +, *, ', 1, 70\rangle$ é uma álgebra booleana;

(*d*) Mostre que $\mathscr{B} = \langle D_{18}, +, *, ', 1, 18\rangle$ *não* é uma álgebra booleana;

(*e*) Generalize os resultados acima e mostre que $\mathscr{B} = \langle D_N, +, *, ', 1, N\rangle$ não é uma álgebra booleana se N possui divisores do tipo p^2 para algum p.

4. Usando somente os axiomas que definem uma álgebra booleana, mostre que valem:

(*a*) As leis distributivas $(a+b)c = ac+bc$ e $ab+c = (a+c)(b+c)$;

(*b*) As leis de De Morgan $(a+b)' = a'b'$ e $(ab)' = a'+b'$.

5. Mostre que as seguintes identidades valem para os elementos de uma álgebra booleana qualquer:

(*a*) $a + a'b = a + b$;

(*b*) $a + ab + b = a + b$;

(*c*) $a + b(a+c) = (a+b)(a+c)$;

(*d*) $a + b + a'b'c = a + b + c$.

II. Expressões booleanas

1. Para cada uma das expressões booleanas a seguir, calcule seu valor nos pontos $(a,b) = (0,0)$, $(0,1)$, $(1,0)$ e $(1,1)$ ou $(a,b,c) = (0,0,0)$, $(0,0,1)$, $\ldots$, $(1,1,1)$:

(*a*) $\alpha(a,b) = (ab+ab')(a'+b)$;

(*b*) $\beta(a,b) = (a+ab+b)(a+b')$;

(*c*) $\gamma(a,b,c) = (a+b+c)(a'+b'+c')$;

(*d*) $\delta(a,b,c) = (a+b'c)(b+c')$.

2. Para cada uma das expressões booleanas α, β, γ e δ definidas no exercício II.1, calcule as seguintes expressões sobre a álgebra booleana $\mathscr{B}$ definida no exercício I.2:

(*a*) $\alpha(1,e)$, $\alpha(a,b)$, $\alpha(f,1)$;

(*b*) $\beta(d,d)$, $\beta(a,b)$, $\beta(b,a)$;

(*c*) $\gamma(1,e,a)$, $\gamma(a,b,e)$, $\gamma(1,f,0)$;

(*d*) $\delta(e,c,1)$, $\delta(1,a,b)$, $\delta(f,d,d)$;

(*e*) $\alpha(e,1)\beta(a,c),\ \gamma(e,a,1)\delta(b,b,c)$.

3. Reduza as seguintes expressões booleanas à sua forma disjuntiva:

   (*a*) $a(ab'+a'b)$;

   (*b*) $a(a'+b')'$;

   (*c*) $(a+b)(b+c)(c+a)$;

   (*d*) $abc(a+b)$;

   (*e*) $(a+b)'(a'b)'$.

4. Simplifique e reduza as seguintes expressões booleanas à sua forma disjuntiva normal:

   (*a*) $(a'+b)'+bc'$;

   (*b*) $(a+b+c)(a'+b+c)$;

   (*c*) $(a+b'+c)(a'+b+c)$;

   (*d*) $b(a+bc')$;

   (*e*) $(a'+b)'(a+b')+ac'$.

5. Reduza as seguintes expressões à sua forma disjuntiva, isto é, como uma união de interseções:

   (*a*) $(\overline{A\cup B})\cap(B\cup\overline{C})$;

   (*b*) $(\overline{B\cap C})\cap(\overline{\overline{A}\cap C})$.

   Note que $\langle\mathscr{P}(S),\cup,\cap,\bar{},\varnothing,S\rangle$ é uma álgebra booleana para um conjunto finito S qualquer.

6. Mostre que valem as seguintes tautologias:

   (*a*) $ab+a'b'\equiv(a\equiv b)$;

   (*b*) $abc+a'bc+ab'c+abc'\equiv(ab+bc+ca)$.

III. Formas disjuntivas mínimas

1. Para cada uma das expressões booleanas α a seguir, verifique se o produto fundamental π dado é um implicante primo de α. *Dica*: você pode fazer isso expressando α e π em sua forma disjuntiva normal e verificando se π corresponde a algum termo de α.

   (*a*) $\alpha=ab'+abc'+a'bc',\ \pi=ac'$;

   (*b*) $\alpha=ab+ab'c'+a'b'c,\ \pi=ab'$;

   (*c*) $\alpha=c+abc+abc',\ \pi=ab$;

(*d*) $\alpha = a'bc + ab'c + abc,\ \pi = abc.$

2. Encontre a forma disjuntiva mínima para as seguintes expressões booleanas usando mapas de Karnaugh e indique o número de formas disjuntivas mínimas diferentes que você consegue encontrar para cada uma delas:

   (*a*) $\alpha = abc + abc' + a'c' + a'b'c + a'bc'$;

   (*b*) $\beta = ab + a'(b' + c') + bc'$;

   (*c*) $\gamma = a + bc + a'b'c'$;

   (*d*) $\delta = abc' + ab'c + ab'c' + a'bc' + a'b'c$;

   (*e*) $\eta = abc + abc' + ab'c' + a'b'c$;

   (*f*) $\mu = abc + abc' + ab'c + ab'c' + a'bc + a'bc' + a'b'c + a'b'c'$.

   (*g*) $\nu = ab + a'bc' + a'b'c$;

   (*h*) $\rho = ac + abc' + a'bc + a'b'c$;

3. Encontre a forma disjuntiva mínima para as seguintes expressões booleanas dadas como soma de mintermos usando mapas de Karnaugh e indique o número de formas disjuntivas mínimas diferentes que você consegue encontrar para cada uma delas:

   (*a*) $\alpha = \sum m(3,6,7,13,14,15)$; (*b*) $\beta = \sum m(3,4,5,6,7,8,12)$;

   (*c*) $\gamma = \sum m(0,1,2,3,4,5,6,7)$; (*d*) $\delta = \sum m(4,6,8,10,13,14)$;

   (*e*) $\eta = \sum m(1,2,3,4,6,9)$; (*f*) $\mu = \sum m(0,1,2,4,8)$.

4. Encontre uma forma disjuntiva mínima para as expressões booleanas representadas pelos seguintes mapas de Karnaugh:

(*a*)

	bc	bc'	$b'c'$	$b'c$
a	✓		✓	✓
a'	✓			✓

(*b*)

	bc	bc'	$b'c'$	$b'c$
a	✓	✓	✓	
a'			✓	✓

(*c*)

	bc	bc'	$b'c'$	$b'c$
a	✓		✓	✓
a'	✓	✓		

(*d*)

	bc	bc'	$b'c'$	$b'c$
a	✓			✓
a'	✓	✓	✓	✓

5. Encontre uma forma disjuntiva mínima para as expressões booleanas representadas pelos seguintes mapas de Karnaugh:

(a)

	cd	cd'	$c'd'$	$c'd$
ab	✓	✓	✓	
ab'			✓	
$a'b'$	✓	✓	✓	✓
$a'b$	✓	✓		

(b)

	cd	cd'	$c'd'$	$c'd$
ab				✓
ab'		✓	✓	
$a'b'$	✓	✓		
$a'b$	✓			

(c)

	cd	cd'	$c'd'$	$c'd$
ab		✓		
ab'		✓	✓	✓
$a'b'$	✓	✓	✓	
$a'b$	✓	✓	✓	

(d)

	cd	cd'	$c'd'$	$c'd$
ab	✓	✓		
ab'	✓	✓	✓	✓
$a'b'$			✓	✓
$a'b$	✓	✓		

IV. Circuitos digitais

1. Desenhe circuitos digitais para cada uma das seguintes expressões booleanas:

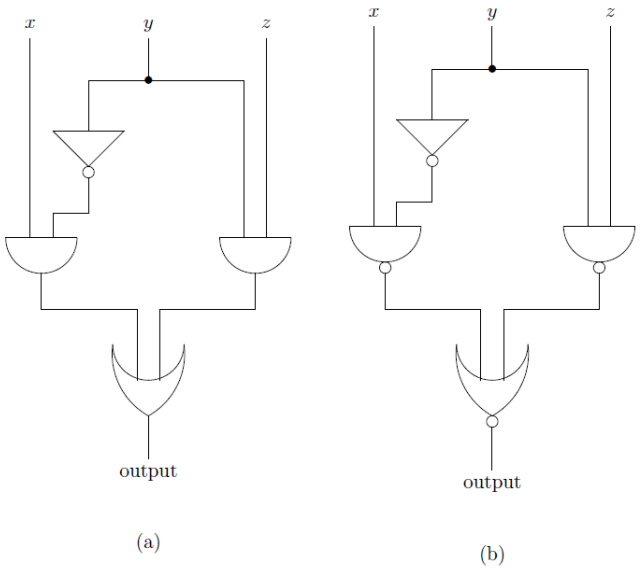
   (*a*) $x+y+z$;

   (*b*) $xy+yz+zx$;

   (*c*) $x+(x'y')(y+z)$.

2. Para cada um dos circuitos na figura abaixo, (*a*) calcule a expressão lógica para cada porta e (*b*) dê a expressão booleana para sua saída (*output*).

3. Desenhe um circuito equivalente ao circuito (*b*) da figura abaixo usando somente portas **and**, **or** e **not**.



Circuitos para os exercícios IV.2 e IV.3.

V. *Divertissement*: Mapas de Karnaugh e o *set cover problem*

Para encontrar a forma disjuntiva mínima de uma expressão booleana usando mapas de Karnaugh devemos encontrar o menor número possível de retângulos fundamentais que conjuntamente incluam a maior quantidade possível de produtos fundamentais. Esse problema é inteiramente análogo ao ***problema do recobrimento de conjuntos*** (***set cover problem***): dada uma coleção $S_1, \ldots, S_n$ de subconjuntos de um conjunto finito S, cada um associado respectivamente a um custo (ou peso) $w_1, \ldots, w_n \geqslant 0$, encontre uma subcoleção $S_{i_1}, \ldots, S_{i_k}$ tal que $S = S_{i_1} \cup \cdots \cup S_{i_k}$ minimiza o custo total $\sum_j w_{i_j}$. Essa subcoleção é chamada "recobrimento" de S. Uma versão simplificada (mas bastante comum e útil) do problema do recobrimento considera todos os custos $w_i = 1$, de forma que o custo total associado ao recobrimento $S = S_{i_1} \cup \cdots \cup S_{i_k}$ é simplesmente dado por k, o número de subconjuntos envolvidos no recobrimento.

Um problema prático relacionado ao problema do recobrimento de conjuntos consiste, por exemplo, em encontrar a menor expressão regular que seleciona determinada lista de substrings mas nenhuma outra (por exemplo, para encontrar o menor conjunto de *labels* que cubra todos os itens de um banco de dados). Em uma outra aplicação, por volta de 1988 pesquisadores do Thomas J. Watson Research Center da IBM em Yorktown Heights, NY, aplicaram o problema para otimizar buscas por vírus de computadores. A partir de 9000 substrings de 20 ou mais bytes consecutivos de código retiradas de cerca de 5000 vírus que não costumam ser encontradas em "código benigno", os pesquisadores concluíram que 180 substrings eram suficientes para cobrir o conjunto das 9000 substrings—isto é, bastaria procurar por 180 substrings para verificar a presença de algum dos vírus de computador usados na pesquisa.

O problema do recobrimento de conjuntos é um problema NP-difícil, de forma que não devemos esperar encontrar um algoritmo eficiente que sempre consiga resovê-lo em tempo polinomial.[(a)] Existem, no entanto, algoritmos aproximados ("heurísticas") que encontram uma subcoleção de conjuntos que recobrem S a um custo muito próximo do mínimo possível. Em 1979, Vašek Chvátal descreveu um algoritmo ganancioso (*greedy*) que encontra uma solução I em tempo polinomial cujo custo é no máximo $H_n \simeq \ln n$ [(b)] maior que o da solução mínima I^*, isto é, $\sum_{i \in I} w_i / \sum_{i \in I^*} w_i \leqslant H_n$.[(c)] O Algoritmo C descreve a heurística de Chvátal para o problema do recobrimento de conjuntos.

No algoritmo de Chvátal, os subconjuntos $S_1, \ldots, S_n$ são escolhidos em uma sequência de rodadas. Em cada rodada, escolhemos o subconjunto S_i disponível que possui a melhor razão

---

[(a)] R. M. Karp, "Reducibility among combinatorial problems", in R. E. Miller, J. W. Thatcher, J. D. Bohlinger (eds.), *Complexity of Computer Computations*, The IBM Research Symposia Series (Boston: Springer, 1972), pp. 85–103.

[(b)] O n-ésimo número harmônico H_n é dado por $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$ e vale aproximadamente $H_n \simeq \ln n + \gamma + \frac{1}{2n}$ para valores grandes de n, onde $\gamma \simeq 0.57721$ é a constante de Euler (às vezes denominada de Euler-Mascheroni).

[(c)] V. Chvatal, "A greedy heuristic for the set-covering problem", *Mathematics of Operations Research*, v. 4, n. 3, pp. 233–235 (1979).

Algorithm C Heurística de Chvátal (1979) para aproximar o recobrimento mínimo de S

Require: Conjuntos $S_1, \ldots, S_n \subseteq S$ tais que $\bigcup_i S_i = S$ e respectivos custos $w_1, \ldots, w_n$

1: $I \leftarrow \varnothing$
2: **while** $S \neq \varnothing$ **do**
3: $\quad i \leftarrow \arg\min_{j:\, S_j \neq \varnothing} \dfrac{w_j}{|S_j|}$
4: $\quad I \leftarrow I \cup \{i\}$
5: $\quad S \leftarrow S \setminus S_i$
6: $\quad$ **for all** $j \notin I$ **do**
7: $\quad\quad S_j \leftarrow S_j \setminus S_i$
8: $\quad$ **end for**
9: **end while**

Ensure: $\{S_i : i \in I\}$ é uma cobertura para S ao custo $w = \sum_{i \in I} w_i$

entre seu custo w_i e número de novos elementos que ele pode acrescentar ao conjunto de cobertura. Em caso de empate, escolhemos um subconjunto qualquer dentre os empatados. Esse algoritmo termina em tempo polinomial, já que não pode haver mais do que n rodadas, cada uma contendo o cálculo de no máximo $O(n)$ razões, cada uma em tempo constante.

Exercício. Simule a execução do Algoritmo C no caso dos subconjuntos $S_1 = \{2,4,5\}$, $S_2 = \{1,4\}$, $S_3 = \{3,4,5\}$, $S_4 = \{2,3,4\}$ e $S_5 = \{3,4\}$ com custos $w_i = |S_i|$. Qual é a solução ótima neste caso?

Exercício. Adapte o Algoritmo C para encontrar formas disjuntivas mínimas de expressões booleanas. Você deve primeiramente reduzir o problema de recobrimento de conjuntos ao problema conhecido como 3-SAT (consulte o artigo de R. M. Karp mencionado anteriormente).

O aluno interessado em álgebras booleanas e suas aplicações ao design de circuitos digitais deve consultar o livro-texto de Thomas L. Floyd, *Digital Fundamentals*, 11a. ed. (Upper Saddle River: Pearson, 2015). Uma abordagem mais matemática do assunto, porém bastante clara e aplicada, com muitos exemplos, aparece na monografia de R. Lidl e G. Pilz, *Applied Abstract Algebra*, 2a. ed. (New York: Springer, 1998). Este último texto contém também material sobre criptografia e códigos corretores de erro. Finalmente, uma discussão lúcida e didática de algoritmos aproximados para problemas NP-difíceis do ponto de vista da pesquisa operacional (programação inteira, otimização combinatória) é dada por D. P. Williamson e D. B. Shmoys, *The Design of Approximation Algorithms* (New York: Cambridge University Press, 2011).

⋆ ⋆ ⋆