EE 044252: Digital Systems and Computer Structure

Spring 2018

Lecture 2: *Switching Algebra and Functions*

EE 044252: Digital Systems and Computer Structure

Topic	wk	Lectures	Tutorials	Workshop	Simulation
Arch	1	Intro. RISC-V architecture	Numbers. Codes		
Comb	2	Switching algebra & functions	Assembly programming		
	3	Combinational logic	Logic minimization	Combinational	
	4	Arithmetic. Memory	Gates		Combinational
Seq	5	Finite state machines	Logic		
	6	Sync FSM	Flip flops, FSM timing	Sequential	Sequential
	7	FSM equiv, scan, pipeline	FSM synthesis		
	8	Serial comm, memory instructions	Serial comm, pipeline		
µArch	9	Function call, single cycle RISC-V	Function call		
	10	Multi-cycle RISC-V	Single cycle RISC-V		Multi-cycle
	11	Interrupts, pipeline RISC-V	Multi-cycle RISC-V		
	12	Dependencies in pipeline RISC-V	Microcode, interrupts		
	13		Depend. in pipeline RISC-V		

Agenda

- Switching Algebra
- Switching Functions

Agenda

- Switching Algebra
- Switching Functions

Switching Algebra אלגברת מיתוג

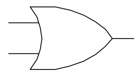
<u>הגדרה</u>

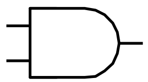
- אלגברת מיתוג:
  - הקבוצה $\{1,0\}$
    - בלוגיקה מתימטית מסכימים שהקבוע 1 מייצג "אמת" והקבוע 0 מייצג "שקר"
  - עליה מוגדרות:
    - שתי פעולות <u>בינריות</u> (הפועלות על שני אופרנדים)
    - פעולה <u>אונרית</u> (הפועלת על אופרנד אחד)

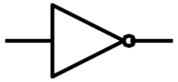
  המוגדרות כלהלן:

אלגברת מיתוג – פעולות

OR (+,|)	AND (·,&)	NOT (` , ¯)
0+0=0	0·0=0	0`=1
0+1=1	0·1=0	1`=0
1+0=1	1·0=0	
1+1=1	1·1=1	







הטבלה קובעת מתי תוצאת הפעולה היא אמת (או שקר)
לכן היא מכונה "טבלת אמת" (truth table)

- קבוצת הערכים סופית (גודל 2)
- לכן קבוצת האפשרויות לבחירת 2 אופרנדים גם היא סופית (גודל $2^2=4$)
- לכן אפשר להגדיר כל פעולה באופן מלא, על ידי מניית כל האפשרויות (בניגוד לאלגברה על מספרים)
- הפעולות באלגברת מיתוג דומות לפעולות חיתוך, איחוד והשלמה בתורת הקבוצות

טבלת האמת

- הטבלאות להלן מכילות משתנים (x, y) אותם נגדיר בהמשך

x	NOT(x)
0	1
1	0

x	y	AND(x,y)	OR(x,y)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

The story of Switching Algebra

- In math (and math logic), **Boolean** algebra is a branch of algebra in which the values of the variables are the truth values true and false (denoted 1 and 0)
- Operations of Boolean algebra are
  - conjunction (and, $\wedge$)
  - disjunction (or, $\vee$)
  - negation (not, $\neg$)
- A formalism for describing logical relations (similar to algebra describing numeric relations)
- Boolean algebra was introduced by George Boole:
  - The Mathematical Analysis of Logic (1847)
  - An Investigation of the Laws of Thought (1854)
- In the 1930s, while studying switching circuits, Claude Shannon observed that one could also apply the rules of Boole's algebra in this setting, and he introduced switching algebra
- “Switching algebra" and "Boolean algebra" are ‘almost’ the same

משתני מיתוג וקבועי מיתוג

- קבועי מיתוג: כל אחד משני הערכים '0' ו- '1'
- נסכים על המשמעות הלוגית
  - 1 הוא אמיתי, 0 הוא שקרי (אלו "ערכי אמת")
  - (truth values) 1 is TRUE, 0 is FALSE
- משתנה מיתוג: משתנה שיכול לקבל אחד משני הערכים '0' או '1'

כאשר $x=1 \Leftrightarrow x\neq0$

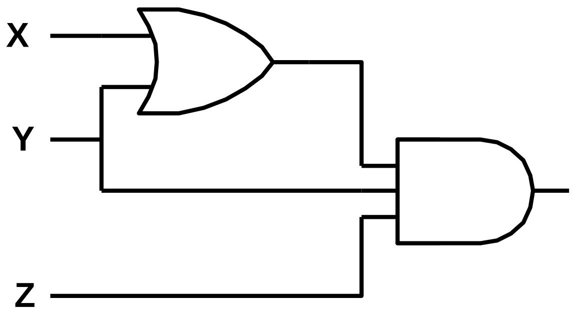
כאשר $x=0 \Leftrightarrow x\neq1$

כמובן, $x=1=TRUE$, ו-$x=0=FALSE$

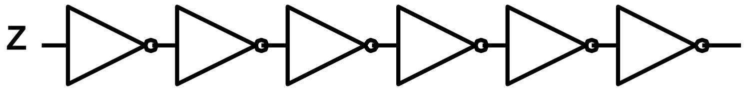
ביטוי מיתוג

- ביטוי-מיתוג (Switching expression) הוא צירוף סופי של
  - משתני מיתוג (למשל x,y,z)
  - קבועי מיתוג (0, 1)
  - פעולות מיתוג
- המוגדר באופן אינדוקטיבי :
  - כל משתנה-מיתוג או קבוע-מיתוג הינו ביטוי מיתוג
  - אם T_1 ו- T_2 הינם ביטויי מיתוג, כך גם T_1+T_2, $T_1 \bullet T_2$ ו- $(T_1)`$
  - צירוף סימנים <u>שאינו</u> מתקבל כך, <u>איננו</u> ביטוי מיתוג

ביטוי מיתוג - דוגמות



- $(x+y) \bullet z \bullet y$



- $z``````$

- $x++y$ אינו ביטוי מיתוג

סדר ביצוע הפעולות הבוליאניות

- לפי הסוגריים
- בהעדר סוגריים:
  - קדימות ראשונה: שלילה (NOT)
  - קדימות שניה: AND
  - קדימות שלישית: OR

ערך (אמת) של בטוי מיתוג תחת השמה

- **<u>הגדרה</u>**

ערך (האמת) של ביטוי מיתוג T תחת השמת ערכי אמת (אמיתי או שקרי) הוא הערך שמקבל הביטוי כאשר מציבים לכל משתנה מיתוג בו את ערך האמת שלו בהשמה

- <u>דוגמה</u>: נחשב את הביטוי $A` + C + B`C + AC`$ כאשר נתון: $A=0, B=0, C=1$

$A` + C + B`C + AC`$

$0' + 1 + 0' \cdot 1 + 0 \cdot 1' =$

$1 + 1 + 1 \cdot 1 + 0 \cdot 0 =$

$1 + 1 + 1 + 0 =$

$1 + 1 + 0 =$

$1 + 0 = 1$

זהות מיתוג

- יהיו T_1 ו- T_2 בטויי מיתוג. השיויון $T_1=T_2$ הינו זהות מיתוג אם תחת כל השמה מקבלים T_1 ו-T_2 ערכים שווים
- דוגמאות:

זהות	לא זהות
x=x	x=y
x•y=y•x	x•y=y+x

- ההוכחה מתבצעת ע"י בדיקת **כל** הצירופים האפשריים של המשתנים
- כלומר, כתיבת טבלת האמת
- בדיקה כזו קרויה אינדוקציה שלמה (perfect induction). מדוע?

זהות מיתוג

- הערה: בזהות מיתוג ניתן להחליף כל משתנה בבטוי מיתוג. למשל:

$$x \cdot y = y \cdot x$$

$$(w+zq) \cdot y = y \cdot (w+zq)$$

זהויות מיתוג בסיסיות (א)

1. אדישות (Idempotence)

$$x+x=x$$

$$x \bullet x=x$$

- הוכחה באמצעות אינדוקציה שלמה
- לפי טבלאות האמת של AND, OR:

$$0+0=0,\ 1+1=1$$

$$0 \bullet 0=0,\ 1 \bullet 1=1$$

- אין עוד אפשרויות, לכן מש״ל

זהויות מיתוג בסיסיות (ב)

2. ערכים אדישים ושולטים

$x+1=1$	1 שולט ב-OR
$x \cdot 0=0$	0 שולט ב-AND
$x+0=x$	0 אדיש ב-OR
$x \cdot 1=x$	1 אדיש ב-AND

זהויות מיתוג בסיסיות (ג)

3. <u>חילוף (קומוטטיביות, Commutativity)</u>

$$x+y=y+x$$
$$x \cdot y=y \cdot x$$

4. <u>קיבוץ (אסוציטיביות, Associativity)</u>

$$(x+y)+z=x+(y+z)$$
$$(x \cdot y) \cdot z=x \cdot (y \cdot z)$$

זהויות מיתוג בסיסיות (ד)

5. <u>השלמה (Complementation)</u>

$$x+x` =1$$

$$x \cdot x` =0$$

הוכחה באמצעות אינדוקציה שלמה וגם על פי ערכים אדישים (זהות 2)

זהויות מיתוג בסיסיות (ה)

6. <u>פילוג (דיסטריבוטיביות, Distributivity)</u>

(א) $x\,(y+z)=xy+xz$ כמו באלגברה רגילה

(ב) $x+(y\cdot z)=(x+y)\cdot(x+z)$ <u>בניגוד</u> לאלגברה רגילה

הוכחת (ב) באמצעות אינדוקציה שלמה:

x	y	z	yz	x+yz	x+y	x+z	(x+y)(x+z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

עקרון הדואליות

- בכל זהות עד כה, אם נחליף

$$AND \leftrightarrow OR$$

$$1 \leftrightarrow 0$$

למשל $(1+A)(B+0)$

$\leftarrow$ $(0 \cdot A)+(B \cdot 1)$

נקבל זהות (חדשה) דואלית

- משפט (עקרון הדואליות): אם מתקיימת הזהות

P(x,y,…,0,1,AND,OR,NOT)= Q(x,y,…,0,1,AND,OR,NOT)

אזי מתקיימת גם הזהות

P(x,y,…,1,0,OR,AND,NOT)= Q(x,y,…,1,0,OR,AND,NOT)

הוכחה: על פי הסימטריה בין טבלאות האמת של AND ושל OR

- מסקנה: מספיק להוכיח תכונה אחת מכל זוג (דואלי), והתכונה השנייה נובעת מעיקרון הדואליות (אכן, כל הזהויות הקודמות נוסחו עבור זוגות דואליים)

שימו לב: ביטויים דואליים אינם זהים זה לזה!

רישום הזהויות כטבלה דואלית

1. אדישות	$x \cdot x = x$	$x+x=x$
2. ערכים אדישים ושולטים	$x \cdot 0 = 0$ $x \cdot 1 = x$	$x+0=x$ $x+1=1$
3. חילוף	$x \cdot y = y \cdot x$	$x+y=y+x$
4. קיבוץ	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x+y)+z=x+(y+z)$
5. השלמה	$x \cdot x` = 0$	$x+x` = 1$
6. פילוג	$x + (y \cdot z) = (x+y) \cdot (x+z)$	$x(y+z)=xy+xz$

זהויות מיתוג בסיסיות (ו)

7. חוק הבליעה (Absorption)

זהויות דואליות

$$x+xy=x$$
$$x(x+y)=x$$

הוכחה באמצעות אינדוקציה שלמה וגם על סמך זהויות קודמות :

1. אדישות	$x \cdot x=x$	$x+x=x$
2. ערכים אדישים ושולטים	$x \cdot 0=0$ $x \cdot 1=x$	$x+0=x$ $x+1=1$
3. חילוף	$x \cdot y=y \cdot x$	$x+y=y+x$
4. קיבוץ	$(x \cdot y) \cdot z=x \cdot (y \cdot z)$	$(x+y)+z=x+(y+z)$
5. השלמה	$x \cdot x`=0$	$x+x`=1$
6. פילוג	$x+(y \cdot z)=(x+y) \cdot (x+z)$	$x(y+z)=xy+xz$

$$x+xy=\overbrace{x \cdot 1}^{(2)}+xy=\overbrace{x(1+y)}^{(6),\,(2),\,(2)}=x$$

$$x(x+y)=\underbrace{(x+0)}_{(2)} \cdot (x+y)=\underbrace{x+0 \cdot y}_{(6),\,(2),\,(2)}=x$$

זהויות מיתוג בסיסיות (ז)

8. חוק הבליעה--המשך (Absorption)

זהויות דואליות

$$\begin{cases} x+x'y=x+y \\ x(x'+y)=xy \end{cases}$$

הוכחה באמצעות אינדוקציה שלמה וגם על סמך זהויות קודמות:

1. אדישות	$x \bullet x=x$	$x+x=x$
2. ערכים אדישים ושולטים	$x \bullet 0=0$ $x \bullet 1=x$	$x+0=x$ $x+1=1$
3. חילוף	$x \bullet y=y \bullet x$	$x+y=y+x$
4. קיבוץ	$(x \bullet y) \bullet z=x \bullet (y \bullet z)$	$(x+y)+z=x+(y+z)$
5. השלמה	$x \bullet x`=0$	$x+x`=1$
6. פילוג	$x+(y \bullet z)=(x+y) \bullet (x+z)$	$x(y+z)=xy+xz$
7. בליעה (ראשון)	$x+xy=x$	$x(x+y)=x$

$$x+x'y=\overbrace{(x+x')\bullet(x+y)}^{(6)}=\overbrace{1\bullet(x+y)}^{(5),(2)}=x+y$$

$$x(x'+y)=\underbrace{xx'+xy}_{(6)}=\underbrace{0+xy=xy}_{(5),(2)}$$

דוגמה נוספת לפישוט ביטוי

$$x`y`z+yz+xz$$

$$= z(x`y`+y+x) \qquad \textbf{(6)}$$

$$= z(x` \quad +y+x) \qquad \textbf{(8)}$$

$$= z \qquad \textbf{(5), (2), (2)}$$

1. אדישות	x•x=x	x+x=x
2. ערכים אדישים ושולטים	x•0=0 x•1=x	x+0=x x+1=1
3. חילוף	x•y=y•x	x+y=y+x
4. קיבוץ	(x • y) • z=x •(y • z)	(x+y)+z=x+(y+z)
5. השלמה	x•x`=0	x+x`=1
6. פילוג	x +(y•z)=(x+y)•(x + z)	x (y+z)=xy+xz
7. בליעה (ראשון)	x+xy=x	x(x+y)=x
8. בליעה (שני)	x+x′y=x+y	x(x′+y)=xy

זהויות מיתוג בסיסיות (ח)

9. חוק הקונסנזוס

$$xy+x`z+yz=xy+x`z$$

הוכחה:

1. אדישות	$x \bullet x=x$	$x+x=x$
2. ערכים אדישים ושולטים	$x \bullet 0=0$ $x \bullet 1=x$	$x+0=x$ $x+1=1$
3. חילוף	$x \bullet y=y \bullet x$	$x+y=y+x$
4. קיבוץ	$(x \bullet y) \bullet z=x \bullet (y \bullet z)$	$(x+y)+z=x+(y+z)$
5. השלמה	$x \bullet x`=0$	$x+x`=1$
6. פילוג	$x+(y \bullet z)=(x+y) \bullet (x+z)$	$x(y+z)=xy+xz$
7. בליעה (ראשון)	$x+xy=x$	$x(x+y)=x$
8. בליעה (שני)	$x+x'y=x+y$	$x(x'+y)=xy$

$$\begin{aligned}
& xy + x`z + yz \\
= \; & xy + x`z + yz \bullet 1 && (2) \\
= \; & xy + x`z + yz(x+x`) && (5) \\
= \; & {\color{red}xy} + {\color{green}x`z} + {\color{red}yzx}+{\color{green}yzx`} && (6) \\
= \; & {\color{red}xy} + {\color{green}x`z} && (7), (7)
\end{aligned}$$

- אין באלגברת מיתוג פעולות הופכיות ל-AND ול-OR (אסור לצמצם)!
- למשל

$$A+B=A+C$$

איננו גורר

$$B=C$$

הפרכה באמצעות דוגמה נגדית. איזו ?

חוקי דה-מורגן (De Morgan)

- חוקים אלו מאפשרים טיפול במשלימים

10. היפוך עצמי (involution)

$$(x`)` = x$$

11. חוקי דה-מורגן

$$(x+y)` = x` \bullet y`$$

$$(xy)` = x` + y`$$

הוכחה באמצעות אינדוקציה שלמה

x	y	x'	y'	x+y	(x+y)′	x'y′
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

$x+x=x$	$x \bullet x=x$	1. אדישות
$x+0=x$ $x+1=1$	$x \bullet 0=0$ $x \bullet 1=x$	2. ערכים אדישים ושולטים
$x+y=y+x$	$x \bullet y=y \bullet x$	3. חילוף
$(x+y)+z=x+(y+z)$	$(x \bullet y) \bullet z=x \bullet (y \bullet z)$	4. קיבוץ
$x+x`=1$	$x \bullet x`=0$	5. השלמה
$x\ (y+z)=xy+xz$	$x+(y \bullet z)=(x+y) \bullet (x+z)$	6. פילוג
$x(x+y)=x$	$x+xy=x$	7. בליעה (ראשון)
$x(x'+y)=xy$	$x+x'y=x+y$	8. בליעה (שני)
	$xy+x`z+yz=xy+x`z$	9. קונצנזוס
	$(x`)`=x$	10. היפוך עצמי
$(x+y)`=x` \bullet y`$	$(xy)`=x`+y`$	11. דה-מורגן

דוגמה לשימוש בחוקי דה-מורגן

10. היפוך עצמי	$(x`)`=x$	
11. דה-מורגן	$(xy)`=x`+y`$	$(x+y)`=x`\cdot y`$

$$(x+y`z)` = (x+(y`z))`$$

$$\underset{(11)}{=} x`(y`z)` = x`((y`)z)`$$

$$\underset{(11)}{=} x`((y`)`+z`) \underset{(10)}{=} x`(y+z`)$$

חוק דה-מורגן המוכלל

באופן כללי

$$P(x,y,\ldots,0,1,AND,OR)` = P(x`,y`,\ldots,1,0,OR,AND)$$

דוגמה:

$$(x`+y+z)` = (x`)`y`z` = xy`z`$$

הערה: נשים לב שפעולות NOT אינן מושפעות מחוקי דה-מורגן

שימו לב: הביטוי המתקבל זה לביטוי <u>ההופכי</u> של הביטוי המקורי!

דוגמה לשימוש בחוק דה-מורגן המוכלל

$(x+y)[x`(y`+z`)]`+x`y`+x`z`=$ } (מוכלל 11), (10)

$=(x+y)(x+yz) +x`y`+x`z`=$ } (6), (1)

$=x+yz+x`y`+x`z`=$ } (8), (8), (8)

$=x+y`+z`+y=$ } (5), (2)

$=1$

(1)	$x+x=x$	$x \bullet x=x$
(2)	$x+0=x$ $x+1=1$	$x \bullet 0=0$ $x \bullet 1=x$
(3)	$x+y=y+x$	$x \bullet y=y \bullet x$
(4)	$(x+y)+z=x+(y+z)$	$(x \bullet y) \bullet z=x \bullet (y \bullet z)$
(5)	$x+x`=1$	$x \bullet x`=0$
(6)	$x(y+z)=xy+xz$	$x+(y \bullet z)=(x+y) \bullet (x+z)$
(7)	$x+xy=x$	$x(x+y)=x$
(8)	$x+x`y=x+y$	$x(x`+y)=xy$
(9)	$xy+x`z+yz=xy+x`z$	
(10)	$(x`)`=x$	
(11)	$(x+y)`=x` \bullet y`$	$(xy)`=x`+y`$

דוגמא נוספת

$$[(x+y)`z`]`$$
$$=(x`y`)`+z$$
$$=(x+y) + z$$

חוק דה-מורגן המוכלל - הוכחה

- הוכחה באינדוקציה על מספר פעולות AND ו-OR
- בסיס (אפס פעולות)

$$(x^{\overset{m}{``\,``}})` = (x`)^{\overset{m}{``\,``}}$$

- לשם פישוט המשך ההוכחה נבטל בביטוי P כפילות של שערי NOT. בבסיס נותרים רק שני מקרים:

$$P(x)=x, \quad \text{ואז} \quad (x)` = (x`)$$

$$P(x)= x`, \quad \text{ואז} \quad (x`)` = (x``) = x$$

- לשם פישוט נוסף, מעתה נגדיר שכל אחד מ- ...x,y, בביטוי P הינו או משתנה או ההפכי שלו (נגדיר ליטרל [literal] ≡ משתנה או הופכי שלו)

חוקי דה-מורגן – המשך הוכחה

- צעד ראשון:

נוכיח שהחוק מתקיים כאשר יש רק פעולת OR או AND אחת. ההוכחה **באינדוקציה שלמה**, ובקיצור:

$P(x_1, x_2,OR,AND)` = (x_1\ OR/AND\ x_2)` = x_1`AND/OR\ x_2` = P\ (x_1`, x_2`,AND,OR)$

על פי דה-מורגן הבסיסי, כאשר x_1, x_2 ליטרלים (יכולים לבטא משתנה או הופכי שלו).

נשים לב שייתכן מקרה נוסף בו הביטוי P עצמו הינו הופכי של ביטוי, כלומר

$P=(x_1\ OR/AND\ x_2)`$

ואז

$P(x_1, x_2,OR,AND)` = [(x_1\ OR/AND\ x_2)`]` = [x_1`AND/OR\ x_2`]` = P\ (x_1`, x_2`,AND,OR)$

על פי דה-מורגן הבסיסי.

חוקי דה-מורגן – המשך הוכחה

- צעד :

נניח שהחוק מתקיים כאשר מספר <u>פעולות</u> OR ו-AND קטן או שווה ל-n ונוכיח עבור מספר <u>פעולות</u> n+1

$$
\begin{aligned}
&P(x_1, x_2,\ldots, x_k,OR,AND)\text{`} = \\
&(S(x_1, x_2,\ldots, x_k,OR,AND) \;\substack{AND\\OR}\; T(x_1, x_2,\ldots, x_k,OR,AND))\text{`} = \\
&(S\text{`}(x_1, x_2,\ldots, x_k,OR,AND) \;\substack{OR\\AND}\; T\text{`}(x_1, x_2,\ldots, x_k,OR,AND)) = \\
&(S\,(x_1\text{`}, x_2\text{`},\ldots,x_k\text{`},AND,OR) \;\substack{OR\\AND}\; T\,(x_1\text{`}, x_2\text{`},\ldots, x_k\text{`},AND,OR)) = \\
&P\,(x_1\text{`}, x_2\text{`},\ldots, x_k\text{`},AND,OR)
\end{aligned}
$$

שוב ייתכן מקרה נוסף בו P עצמו הינו NOT של ביטוי, הטיפול בכך כמו בשקף הקודם

חוקי דה-מורגן – המשך הוכחה

- צעד במקרה בו P עצמו הינו NOT של ביטוי:

$$P(x_1, x_2,\ldots, x_k,OR,AND)\text{`} =$$
$$[(S(x_1, x_2,\ldots, x_k,OR,AND) \substack{AND \\ OR} T(x_1, x_2,\ldots, x_k,OR,AND))\text{`}]\text{`} =$$
$$[(S\text{`}(x_1, x_2,\ldots, x_k,OR,AND) \substack{OR \\ AND} T\text{`}(x_1, x_2,\ldots, x_k,OR,AND))]\text{`} =$$
$$[(S\ (x_1\text{`}, x_2\text{`},\ldots,x_k\text{`}AND,OR) \substack{OR \\ AND} T\ (x_1\text{`}, x_2\text{`},\ldots, x_k\text{`},AND,OR))]\text{`} =$$
$$P\ (x_1\text{`}, x_2\text{`},\ldots, x_k\text{`},AND,OR)$$

סיכום ביניים

- אלגברת מיתוג (מקרה פרטי של אלגברה בוליאנית) מטפלת בקבוצה סופית של ערכים (0,1 או אמיתי, שקרי) ובשלוש פעולות (AND, OR, NOT)
- מתוך ההגדרות גזרנו את כל הזהויות והוכחנו את נכונותן באמצעות אינדוקציה שלמה או באמצעות זהויות אחרות

Agenda

- Switching Algebra
- Switching Functions

הגדרה

- פונקציית מיתוג של n משתנים

$$f(x_1,\ldots,x_n)$$

הינה כלל התאמה בין 2^n הצירופים השונים של המשתנים $x_1,\ldots,x_n$ לבין הקבוצה $\{0,1\}$

טבלת האמת

- ניתן לתאר פונקציית מיתוג באמצעות טבלת אמת:

$x_1,\ldots,x_n$	$f(x_1,\ldots,x_n)$
0,0,…,0	f(0,0,…,0)
0,0,…,1	f(0,0,…,1)
…	…
1,1,…,1	f(1,1,…,1)

מהו מספר פונקציות המיתוג השונות ב-n משתנים?

$x_1,\ldots,x_n$	$f_0(x_1,\ldots,x_n)$	$f_1(x_1,\ldots,x_n)$	$f_2(x_1,\ldots,x_n)$	…	…
0,0,…,0	$f_0(0,0,\ldots,0)$	$f_1(0,0,\ldots,0)$	$f_2(0,0,\ldots,0)$		
0,0,…,1	$f_0(0,0,\ldots,1)$	$f_1(0,0,\ldots,1)$	$f_2(0,0,\ldots,1)$		
…	…	…	…		
1,1,…,1	$f_0(1,1,\ldots,1)$	$f_1(1,1,\ldots,1)$	$f_2(1,1,\ldots,1)$		

מספר הפונקציות = מספר הווקטורים בגודל 2^n = מספר תת-הקבוצות האפשריות של 2^n שורות

תשובה: 2^{2^n}

n=2 דוגמה

x_1,x_2	$f_0(x_1,x_2)$	$f_1(x_1,x_2)$	$f_2(x_1,x_2)$	…	…
0,0	0	1	0		
0,1	0	0	1		
1,0	0	0	0		
1,1	0	0	0		

$$2^{2^n} = 16$$

דוגמה n=2

X_1,X_2	0	NOR	$X_1' X_2$	X_1'	$X_1 X_2'$	X_2'	XOR	NAND	AND	XNOR	X_2	$X_1'+X_2$	X_1	X_1+X_2'	OR	1
0,0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0,1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1,0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1,1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

צורות קנוניות של פונקציות

- פונקציה אחת ניתנת לתיאור באמצעות ביטויי-מיתוג שונים
- נגדיר שתי צורות סטנדרטיות (קנוניות) לתיאור פונקציות:
  - צורה קנונית של **סכום מכפלות** (Sum of Products, SoP)
    - כגון: $f(x,y,z)=x'y'z+x'yz'+xyz$
  - צורה קנונית של **מכפלת סכומים** (Product of Sums, PoS)
    - כגון: $f(x,y,z)=(x+y+z)(x+y'+z)(x'+y+z')$
- הצורה הקנונית הינה **יחידה** (עד כדי חילוף)
  - לפונקציות בעלות אותה טבלת-אמת, אותה צורה קנונית
  - אלו פונקציות שוות/זהות/שקולות

צורה קנונית של סכום מכפלות

- **הגדרה** : מכפלה המכילה את כל n המשתנים (ליטרלים) תקרא minterm
- כל מכפלה (minterm) מקבלת ערך '1' רק עבור צירוף אחד של ערכי המשתנים
- ערך הביטוי יהיה '1' אם לפחות אחת המכפלות ערכה '1'
- מסקנה :
  - בצורה הקנונית SoP יש לרשום את סכום המכפלות (minterms) המתאימות לשורות בטבלת האמת בהן הפונקציה מקבלת ערך 1

שימו לב: צורה קנונית אינה מתימרת להיות מיטבית. יתרונה בכך שהיא "סטנדרטית", וקל להגיע אליה מכל צורת ביטוי נתונה. לעתים נוח למטרות מימוש, ותמיד מאפשר השוואה קלה בין ביטויים כדי לבדוק אם הם זהים זה לזה.

דוגמה

$f(x,y,z) = \boxed{x'y'z'} + \boxed{xy'z'}$

Decimal code	x,y,z	f(x,y,z)
0	0 0 0	1
1	0 0 1	0
2	0 1 0	0
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	0
7	1 1 1	0

Decimal code	x,y,z	f(x,y,z)
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

$$f\left(x,y,z\right)=\overset{0\ \ 0\ \ 0}{x'y'z'}+\overset{0\ \ 1\ \ 0}{x'yz'}+\overset{0\ \ 1\ \ 1}{x'yz}+\overset{1\ \ 1\ \ 0}{xyz'}+\overset{1\ \ 1\ \ 1}{xyz}$$

צורה קנונית של סכום מכפלות

- SoP מכונה גם Disjunctive Normal Form (DNF)
- רישום מקוצר לפונקציה (הנובע מהצורה הקנונית):

$$f(x,y,z)=\sum(0,2,3,6,7)$$

Decimal code	x,y,z	f(x,y,z)
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

צורה קנונית של מכפלת סכומים (PoS)

- מספיק שאחד הסכומים במכפלת סכומים ישווה ל-'0' בכדי שכל הביטוי ישווה ל-'0'
- maxterm – סכום של n ליטרלים, המתאימים לכל n המשתנים השונים
- maxterm (כפונקציה) מקבל את הערך 0 אך ורק עבור צירוף יחיד של ערכי משתני כניסה

צורה קנונית של מכפלת סכומים (PoS)

- מתקבלת כמכפלה המקסטרמים המתאימים לשורות בטבלת האמת של f() בהן הפונקציה מקבלת ערך 0
- מכונה גם Conjunctive Normal Form (CNF)
- רישומה המקוצר:

$$f(x,y,z)=\Pi(1,4,5)$$

Decimal code	x,y,z	f(x,y,z)
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

Decimal code	x,y,z	f(x,y,z)
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

$$f(x,y,z)=\left(\overset{0}{x}+\overset{0}{y}+\overset{1}{z'}\right)\left(\overset{1}{x'}+\overset{0}{y}+\overset{0}{z}\right)\left(\overset{1}{x'}+\overset{0}{y}+\overset{1}{z'}\right)$$

הקשר בין DNF לבין CNF

- באמצעות חוקי דה-מורגן קל להפיק את סכום המכפלות של f'() מתוך מכפלת הסכומים:

$$f(x,y,z)=\sum(0,2,3,6,7)=x'y'z'+x'yz'+x'yz+xyz'+xyz \qquad DNF$$

$$=\Pi(1,4,5)=(x+y+z')(x'+y+z)(x'+y+z') \qquad CNF$$

$$f(xyz)=f''(xyz)=(x+y+z')(x'+y+z)(x'+y+z')$$

$$\overline{f(x,y,z)}=x'y'z+xy'z'+xy'z$$

$$f(x,y,z) = \sum(0,2,3,6,7) = \Pi(1,4,5)$$
$$\overline{f(x,y,z)} = \sum(1,4,5) = \Pi(0,2,3,6,7)$$

העברת ביטוי-מיתוג לצורה קנונית מבלי להיעזר בטבלת האמת

- נתון ביטוי f בצורת סכום מכפלות (לאו-דווקא קנוני)

1. בדוק כל מכפלה בביטוי: אם היא מינטרם, עבור למכפלה הבאה
2. אחרת:

   2.1. הרחבה: כפול את המכפלה ב-$\left(x_i + x'_i\right)$ עבור כל משתנה x_i החסר במכפלה

   2.2. פתח סוגריים ובטל מכפלות חוזרות

דוגמה

$$f(x,y,z) = x'y + z' + xyz$$

$$= x'y(z+z') + z'(x+x')(y+y') + xyz$$

$$= x'yz + \underline{x'yz'} + z'xy + \underline{z'x'y} + z'xy' + z'x'y' + xyz$$

$$= x'yz + x'yz' + xyz' + xy'z' + x'y'z' + xyz$$

$$= x'y'z' + x'yz' + x'yz + xy'z' + xyz' + xyz$$

$$= \sum(0,2,3,4,6,7)$$

$$= \prod(1,5)$$

$$= (x+y+z')(x'+y+z')$$

דוגמה (המשך)

x y z	f(x,y,z)
0 0 0	1
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	1

וניתן גם לבדוק ישירות בטבלת האמת:

$$f(x,y,z) = x'y + z' + xyz$$

$$f(x,y,z) = \sum(0,2,3,4,6,7)$$
$$= \prod(1,5)$$

דוגמה : מכפלת סכומים

$$g(x,y,z)$$
$$= x'(y'+z)$$
$$= (x'+yy'+zz')\,(y'+z+xx')$$
$$= (x'+y+z)(x'+y+z')\underline{(x'+y'+z)}\,(x'+y'+z')(y'+z+x)\underline{(y'+z+x')}\Big\}\ \textbf{(6)}$$
$$= (x'+y+z)(x'+y+z')(x'+y'+z)(x'+y'+z')(x+y'+z)$$
$$= (x+y'+z)(x'+y+z)(x'+y+z')(x'+y'+z)(x'+y'+z')$$
$$= \prod(2,4,5,6,7)$$
$$= \sum(0,1,3)$$
$$= x'y'z'+x'y'z+x'yz$$

תזכורת : חוק הפילוג

$x+(y\cdot z)=(x+y)\cdot(x+z)$ **(6)**

משפט הפיתוח של Shannon
(יסייע לנו בפיתוח הצורות הקנוניות)

- Boole's / Shannon expansion theorem is the identity:

$$f = x \cdot f(x=1) + x' \cdot f(x=0)$$

- בהרחבה, כל פונקצית מיתוג $f(x_1,\ldots,x_n)$ ניתנת לרישום בתור:

$$\boxed{f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + x'_1 \cdot f(0, x_2, \ldots, x_n)}$$

או:

$$\boxed{f(x_1, x_2, \ldots, x_n) = \left(x_1 + f(0, x_2, \ldots, x_n)\right) \cdot \left(x'_1 + f(1, x_2, \ldots, x_n)\right)}$$

משפט הפיתוח של Shannon

- <u>הוכחה</u>: באינדוקציה שלמה על x_1
- השוויונות מתקיימים בהצבת $x_1=0$
ובהצבת $x_1=1$

$$\boxed{f(x_1,x_2,\ldots,x_n)=x_1\cdot f(1,x_2,\ldots,x_n)+x'_1\cdot f(0,x_2,\ldots,x_n)}$$

$$\boxed{f(0,x_2,\ldots,x_n)=0\cdot f(1,x_2,\ldots,x_n)+1\cdot f(0,x_2,\ldots,x_n)}$$

$$\boxed{f(1,x_2,\ldots,x_n)=1\cdot f(1,x_2,\ldots,x_n)+0\cdot f(0,x_2,\ldots,x_n)}$$

משפט הפיתוח של Shannon

$$\boxed{f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + x'_1 \cdot f(0, x_2, \ldots, x_n)}$$

- באמצעות משפט הפיתוח נקבל:

$$f(x_1, \ldots, x_n) = x_1[x_2 \cdot f(1,1, x_3, \ldots, x_n) + x'_2 \cdot f(1,0, x_3, \ldots, x_n)]$$
$$+ x'_1[x_2 \cdot f(0,1, x_3, \ldots, x_n) + x'_2 \cdot f(0,0, x_3, \ldots, x_n)]$$

$$= x_1 x_2 f(1,1, x_3, \ldots, x_n)$$
$$+ x_1 x'_2 \; f(1,0, x_3, \ldots, x_n)$$
$$+ x'_1 \; x_2 f(0,1, x_3, \ldots, x_n)$$
$$+ x'_1 \; x'_2 \; f(0,0, x_3, \ldots, x_n)$$

מסקנה1: קיבלנו את הצורה הקנונית של סכום המכפלות

$$= \ldots =$$
$$x_1 x_2, \ldots, x_n \; f(1,1, \ldots, 1) \; + \ldots +$$
$$x'_1 \; x'_2 \; , \ldots, x'_n \; f(0,0, \ldots, 0)$$

מסקנה2: לכל פונקציה יש צורה קנונית

רישום קנוני של פונקציות

- משפט: כל פונקציה ניתנת לרישום בצורה הקנונית:

$$\boxed{f\left(x_1,\ldots,x_n\right)=a_o x'_1\, x'_2\,,\ldots,x'_n+\ldots..+a_{2^n-1}\, x_1,\ldots,x_n}$$

- הוכחה, בעזרת משפט הפיתוח של שנון
- משפעות: לכל פונקציה f ב-n משתנים מתאים וקטור $\left(a_o\,,a_1\,,\ldots,a_{2^n-1}\right)$ ומתקיים $a_i = 1$ אם״ם בשורה המתאימה בטבלת האמת יש '1'

(זהו ניסוח מחדש של הניתוח בשקפים 41-43)

פונקציות במשתנה אחד

- עבור משתנה אחד נקבל, ממשפט הפיתוח:

$$\boxed{f(x) = a_o x' + a_1 x}$$

- סך הכל $4 = 2^2$ פונקציות:
  - פונקציית הזהות
  - הפונקציה הקבועה 1
  - הפונקציה הקבועה 0
  - NOT

פונקציות בשני משתנים (תזכורת)

X_1,X_2	0	NOR	$X_1' X_2$	X_1'	$X_1 X_2'$	X_2'	XOR	NAND	AND	XNOR	X_2	$X_1'+X_2$	X_1	X_1+X_2'	OR	1
0,0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0,1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1,0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1,1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

תכונות הפונקציה XOR ‏(Exclusive OR)

- חיבור מודולו 2 : שארית החיבור לאחר חלוקה ב-2

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

- $x \oplus y = 1$ אמ"ם x=1 או y=1, אבל לא שניהם

תכונות הפונקציה XOR

$$A \oplus B = B \oplus A$$

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

$$A(B \oplus C) = (AB) \oplus (AC)$$

$$A = B \quad \Leftrightarrow \quad A \oplus B = 0$$

$$A \oplus 0 = A \qquad A \oplus A = 0$$

$$A \oplus 1 = A' \qquad A \oplus A' = 1$$

$$\overline{A \oplus B} = A \equiv B = EQ(A,B)$$

מערכת פעולות שלמה

- Functionally Complete Operations
- הגדרה: קבוצה של אופרטורים וקבועים נקראת שלמה, אם ניתן לתאר באמצעותה כל פונקצית מיתוג
- דוגמה:
  - הקבוצה $\{\cdot\,,+,'\}$ שלמה (הנחה סמויה – קיימת פונקציית שכפול)
  - לכן גם הקבוצות $\{\cdot,'\}$; $\{+,'\}$ שלמות (נובע מחוקי דה-מורגן)
  - לכן גם כל קבוצה הממששת $\{\cdot,'\}$ או $\{+,'\}$ הינה שלמה

מערכת פעולות שלמה (המשך)

- האופרטור NOR ($\downarrow$) מהווה קבוצה שלמה
  - אופרטור יחיד המהווה קבוצה שלמה קרוי אוניברסלי
- נראה שניתן לממש באמצעותו מערכת שלמה אחרת:

$$x \downarrow y = (x+y)'$$

$$x \downarrow x = (x+x)' = x'$$

הערה: ארגומנט x שוכפל

$$(x \downarrow y) \downarrow (x \downarrow y) = (x \downarrow y)' = (x+y)'' = x+y$$

- באופן דומה, גם NAND מהווה קבוצה שלמה ולכן גם הוא אוניברסלי

למען השלמות: מערכת לא שלמה

- איך מראים שמערכת פעולות S אינה שלמה?
  - מראים פונקציה שלא ניתן לממש בעזרתה. כיצד?

1. מראים ע״י בדיקה של כל אפשרויות הרכבת הפונקציות מ- S שפונקציה מסוימת אינה ניתנת לתיאור (למשל- NOT או AND)
2. מוצאים תכונה P שכל הרכבה של פונקציות מ- S מקיימת אך קיימת פונקציה שאינה מקיימת תכונה זו (למשל ליניאריות)
3. מוכיחים שמספר האפשרויות להרכבת פונקציות מ-S קטן ממספר הפונקציות האפשריות (שיקולי ספירה)

- דוגמאות: {XOR}, {XOR,AND}, {XOR,1},...

x	y	XOR(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

מערכת לא שלמה—דוגמאות

- האם {S = {XOR היא מערכת שלמה? למשל, האם ניתן לממש NAND באמצעותה? NOT ?

$$a\oplus a=0 \quad ; \quad a\oplus 0=0\oplus a=a$$

- האם {S = {XOR, AND היא מערכת שלמה?

  - $a\oplus a=0 \quad ; \quad a\cdot a=a$
  - $a\oplus 0=0\oplus a=a$

    $a\cdot 0=0\cdot a=0$

סיכום

- אלגברת מיתוג (מקרה פרטי של אלגברה בוליאנית) מטפלת בקבוצה סופית של ערכים (0,1 או אמיתי, שקרי) ובשלוש פעולות (AND, OR, NOT)
- מתוך ההגדרות גזרנו את כל הזהויות והוכחנו את נכונותן באמצעות אינדוקציה שלמה או באמצעות זהויות אחרות
- קיימות 2^{2^n} פונקציות אפשריות על n משתני מיתוג
- הצורות הקנוניות SoP (DNF), PoS (CNF) הינן יחידות
- משפט ההרחבה של שנון / בול מסייע לבנות פונקציות בצורות קנוניות
- בעזרת מערכת פעולות שלמה ניתן לייצר כל פונקצית מיתוג