БЕОГРАДСКИ УНИВЕРЗИТЕТ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Мастер рад

Афина еквивалентност Булових функција и примене

ментор:
проф. др Миодраг Живковић

студент:
Милош Ристић
бр. индекса: 1018/2011

комисија:
проф. др Миодраг Живковић
проф. Предраг Јаничић
проф. Филип Марић

Октобар, 2013.

Садржај

Сажетак

У овом раду дат је преглед инваријанти које се могу искористити за утврђивање афине еквивалентности Булових функција и представљен је један алгоритам за утврђивање афине еквивалентности двеју задатих Булових функција заснован на инваријантама. Описана је програмска реализација алгоритма и спровођење тестова за различите класе Булових функција. На крају је дат преглед резултата тестирања.

Кључне речи: Булове функције, афина еквивалентност, инваријанте

Глава 1

Увод

Булове функције имају широку примену у науци. Користе се и у инжењерству за пројектовање логичких кола. Уколико је логичко коло добро испројектовано, преко њега се могу реализовати све Булове функције које су афино еквивалентне са Буловом функцијом коју логичко коло представља. Булове функције су посебно проучаване током шездесетих година прошлог века за потребе пројектовања логичких кола.

Булове функције најширу примену имају у криптографији. Користе се за реализацију блоковских и проточних шифара. Са развојем криптографије наставља се проучавање Булових функција и афине еквивалентности, јер се афино еквивалентне функције понашају слично у криптографским системима. Уколико се систем заснива на функцији за коју је лако пронаћи афино еквивалентну функцију, онда је тај систем лак за разбијање.

Сличност афино еквивалентних Булових функција је некада пожељна, а некада непожељна особина. Зато је потребно посветити се проучавању сличности.

У овом раду представљен је један алгоритам за утврђивање афине еквивалентности двеју задатих Булових функција заснован на инваријантама. У другом поглављу рада дат је преглед математичких појмова и дефиниција потребних за проучавање афине еквивалентности Булових функција. У трећем поглављу представљен је алгоритам за утврђивање афине еквивалентности двеју задатих Булових функција и размотрена је ефикасност представљеног алгоритма. У четвртом поглављу представљена је програмска реализација алгоритма и тестова у програмском језику *C++*. На крају су дати преглед и дискусија резултата тестирања.

Глава 2

Булове функције и афина еквивалентност

У овој глави представљене су неке од основних трансформација Булових функција. Уз помоћ ових трансформација добијају се занимљиве инваријанте различитих скупова Булових функција. Најважнија од трансформација је Волшова трансформација[1] која представља уопштени случај Фуријеове трансформације. Волшова трансформација чини израчунавње многих особина Булових функција знатно лакшим. Да би се увеле дефиниције Булове функције, њених трансформација и инваријанти скупова Булових функција потребно је дати преглед основних марематичких појмова на којима су оне засноване.

2.1 Уводни математички појмови и дефиниције

У овој тачки дат је преглед познатих математичких појмова на основу којих се дефинишу Булове функције и афина пресликавања. Поред ових појмова дате су и дефиниције различитих операција над векторима које се појављују у другим тачкама рада. Детаљан опис појмова може се наћи у било ком уџбенику математике. Овде је само дат преглед оних који су неопходни за даљи рад.

$F_2 = (\{0,1\}, \oplus, *)$ је коначно поље над скупом $\{0,1\}$ са операцијама сабирања и множења. Операције су дефинисане следећим Кејлијевим *(Arthur Cayley)* табелама:

[1] Thomas Cusick, Pantelimon Stanica. *Cryptographic Boolean Functions and Applications*, Academic Press, 2009: 7-22

$\oplus$	0	1
0	0	1
1	1	0

$*$	0	1
0	0	0
1	0	1

$F_2^n = (\mathrm{X}, +, *)$ је векторски простор димензије n над пољем F_2. X је скуп свих $n-$торки облика $x = (x_1, x_2, \ldots, x_n)$, где су сви $x_i \in F_2, i = \overline{1,n}$. Операција сабирања дефинисана је изразом

$$x + y = (x_1 \oplus y_1, x_2 \oplus y_2, x_3 \oplus y_3, \ldots, x_n \oplus y_n) \qquad \forall x, y \in F_2^n$$

Операција множења скаларом дефинисана је изразом

$$a * x = (a * x_1, a * x_2, a * x_3, ..., a * x_n) \qquad \forall x \in F_2^n \forall a \in F_2$$

.

Овде су наведене још две операције у скупу вектора F_2^n. Потребне су због увођења појма афиних пресликавања.
Нека $x \bullet y$ означава скаларни производ два вектора $x, y \in F_2^n$ дат изразом $x \bullet y = x_1 * y_1 \oplus x_2 * y_2 \oplus x_3 * y_3 \oplus \ldots \oplus x_n * y_n$.
Нека $x \cdot y$ означава производ два вектора $x, y \in F_2^n$ члан по члан дат изразом $x \cdot y = (x_1 \cdot y_1, x_2 \cdot y_2, x_3 \cdot y_3, \ldots x_n \cdot y_n)$.

Овде је наведена група регуларних матрица потребна за дефинисање афиних пресликавања Булових функција.
Нека $GL(n, 2)$ означава линеарну групу инвертибилних матрица реда n чији су елементи из F_2. Операције групе $GL(n, 2)$ дефинисане су следећим изразима

$A * B = \sum_{i,j,k=\overline{1,n}} a_{i,k} * b_{k,j}$, где су $a_{i,k}$ и $b_{k,j}$ елементи матрица A и B.

A^{-1} је инверзна матрица матрице A и важи $A * A^{-1} = I$, где је I јединична матрица.
Транспоновање је згодна операција приликом рада са матрицама. Операција транспоновања пресдтавља замену места врста и колона матрице. Нека је са A^T означена транспонована матрица A. Транспоновање се формално се може записати у облику $a_{ij}^T = a_{ji}, i, j = \overline{1,n}$, где су a_{ij}^T и a_{ji} елементи матрица A^T и A.
Још једна згодна операција са матрицама је Кронекеров *(Leopold Kronecker)* производ. Кронекеров производ две матрице дефинисан је изразом

$$A \otimes B = \begin{bmatrix} a_{1,1} * B & \cdots & a_{1,n} * B \\ \vdots & \ddots & \vdots \\ a_{n,1} * B & \cdots & a_{n,n} * B \end{bmatrix}$$

Нека је са $AGL(n,2)$ означена афина група $\{(A,b)|A \in GL(n,2), b \in F_2^n\}$. Операције групе (A,b) дефинисане су следећим изразима:
$(A,u) * (B,w) = (A * B, w * A + u)$
$(A,u)^{-1} = (A^{-1}, u * A^{-1})$
где су $(A,u),(B,w) \in AGL(n,2)$.

Користећи претходне појмове уводе се дефиниције Булове функције и неких специјалних Булових функција.

Дефиниција 1 *Пресликавање $f : F_2^n \to F_2$ је Булова функција од n променљивих.*

Дефиниција 2 *Број променљивих које Булова функција прихвата на улазу назива се арност функције.*

Дефиниција 3 *Афина Булова функција је Булова финкција облика*

$$f(x) = x \bullet a \oplus c$$

за свако $x, a \in F_2^n, c \in F_2$.

Дефиниција 4 *Линеарна Булова функција је специјалан случај афине Булове финкције за $c = 0$. Линеарна Булова функција је Булова функција облика $f(x) = x \bullet a$ за свако $x, a \in F_2^n$.*

Дефиниција 5 *Комплементарна функција Булове функције $f \in p_n$ дефинише се изразом $\overline{f}(x) = 1 \oplus f(x)$.*

Овде се уводе ознаке скупова Булових функција одређене арности. У даљем раду детаљније ће се разматрти инваријанте скупова Булових функција.
Нека је са p_n означен скуп свих полинома из

$$F_2[x_1, x_2, \ldots, x_n]/(x_1^2 = x_1, \ldots, x_n^2 = x_n)$$

Ови полиноми представљају скуп свих Булових функција арности n.
Нека је са $|p_n|$ означен број чланова скупа p_n. Број чланова скупа p_n је 2^{2^n}.
Нека је са A_n означен скуп свих афиних Булових функција арности n. Формално записано $A_n = f(x)|f(x) = x \bullet a \oplus c, x, a \in F_2^n, c \in F_2$.

2.2 Начини представљања Булових функција

Булове функције могу се представити на више начина. Како је тема овог рада усмерена ка криптографији, најпогодније је представити Булових функције у светлу те науке. Најчешћи начини представљања су:

1. Истинитосне таблице и поларне истинитосне таблице
2. Алгебарска нормална форма (АНФ)

2.2.1 Истинитосне таблице и поларне истинитосне таблице

Обзиром да Булове функције раде са дискретним вредностима, природан начина задавања су таблице које садрже све могуће варијанте улаза и одговарајуће вредности излаза за те улазе. Булова функција $f : F_2^n \to F_2$ може се јединствено представити истинитосном таблицом. Истинитосна таблица је вектор

$$f(x) = (f(x_1), f(x_2), f(x_3), \ldots, f(x_{2^n}))$$

где су $x_1, x_2, x_3, \ldots, x_{2^n}$ вектори из F_2^n у лексикографском поретку. Слично, поларна истинитосна таблица представља се вектором

$$\widehat{f}(x) = ((-1)^{f(x_1)}, (-1)^{f(x_2)}, (-1)^{f(x_3)}, \ldots, (-1)^{f(x_{2^n})})$$

где су $x_1, x_2, x_3, \ldots, x_{2^n}$ вектори из F_2^n у лексикографском поретку. Пример истинитосне и поларне истинитосне таблице илустрован је табелом 2.1.

Поларна истинитосна таблица повезана је са истинитосном таблицом везом $\widehat{f}(x) = 1 - 2 * f(x)$, где су $-$ и $*$ операције одузимања и множења у скупу целих бројева.

У наставку наведена су нека важна својства Булових функција везана за истинитосне таблице.

Дефиниција 6 *Хемингова тежина вектора $x \in F_2^n$, у ознаци $wt(x)$, је број јединица у вектору x.*

Дефиниција 7 *Хемингова тежина функције $f \in p_n$, у ознаци $wt(f)$, је број јединица у вектору $f(x)$.*

Дефиниција 8 *Хемингово растојање функција $f, g \in p_n$, у ознаци $d(f, g)$, дефинише се изразом $d(f, g) = wt(f(x) \oplus g(x))$*

Табела 2.1: Пример истинитосне и поларне истинитосне таблице за функцију арности 3

$x \in F_2^n$	$f(x)$	$\widehat{f}(x)$
000	0	1
001	1	−1
010	0	1
011	1	−1
100	0	1
101	1	−1
110	1	−1
111	0	1

Дефиниција 9 *Нелинеарност функције $f \in p_n$, у ознаци N_f, дефинише изразом* $N_f = \min_{\varphi \in A_n} d(f, \varphi)$

Теорема 10 *Булова функција $f \in p_n$ је уравнотежена ако је њена Хемингова тежина 2^{n-1}.*

Доказ. Како је број елемената у $f(x)$ једнак 2^n, а број јединица је 2^{n-1}, следи да је број нула $2n - 2^{n-1} = 2^{n-1}$. ■

Теорема 11 *Нека важна својства Хемингове тежине и растојања су:*

1. $wt(x + y) = wt(x) + wt(y) - 2 * wt(x * y) \qquad \forall x, y \in F_2^n$
2. $d(f, g) = |x \in F_2^n : f(x) \neq g(x)| \qquad \forall f, g \in p_n$
3. $d(f, g) + d(g, h) \geq d(f, h) \qquad \forall f, g, h \in p_n$
4. $d(f, \overline{g}) = 2^n - d(f, g) \qquad \forall f, g \in p_n$

Доказ.

1. Вектор $x * y$ има јединицу на месту i ако оба вектора x и y имају јединицу на месту i. Вектор $x + y$ има јединицу на месту i ако бар један од вектора x и y нема јединицу на месту i. Дакле од укупног броја јединица у векторима x и y треба одузети по две јединице за сваку позицију где оба вектора x и y имају јединицу.

2. Вектор $f(x) \oplus g(x)$ има јединицу на месту где је $f(x) \neq g(x)$. Број вектора x за које је ово испуњено је управо $d(f, g)$.

Табела 2.2: Израчухнавање коефицјената полинома за $n = 3$

улаз X	израз за $f(x)$	коефицјент c_i
000	c_{000}	$c_{000} = f(000)$
001	$c_{000} + c_{001}$	$c_{001} = c_{000} + f(001)$
010	$c_{000} + c_{010}$	$c_{010} = c_{000} + f(010)$
011	$c_{000} + c_{001} + c_{010} + c_{011}$	$c_{011} = c_{000} + c_{001} + c_{010} + f(010)$
100	$c_{000} + c_{100}$	$c_{100} = c_{000} + f(100)$
101	$c_{000} + c_{100} + c_{001} + c_{101}$	$c_{101} = c_{000} + c_{100} + c_{001} + f(101)$
110	$c_{000} + c_{100} + c_{010} + c_{110}$	$c_{110} = c_{000} + c_{100} + c_{010} + f(110)$
111	$c_{000} + c_{001} + c_{010} + c_{011}$ $+c_{100} + c_{101} + c_{110} + c_{111}$	$c_{111} = c_{000} + c_{001} + c_{010} + c_{011}$ $+c_{100} + c_{101} + c_{110} + f(111)$

3. ППС. Нека за неку функцију g важи $d(f,g)+d(g,h) < d(f,h)$. Тада би за $f = h$ било $d(f,g) + d(g,f) < d(f,f)$ што је немогуће.

4. Довољно је показати да су функције $f(x) \oplus g(x)$ и $f(x) \ominus \overline{g}(x)$ комплементарне, а то важи јер је $f(x) \oplus g(x) \oplus f(x) \oplus \overline{g}(x) = 1$.

■

2.2.2 Алгебарска нормална форма

За детаљније проучавање Булових функција и њихових алгебарских својстава погодно је представити Булове функције у облику полинома. Један начин да се то уради је алгебарска нормална форма[5]. Булова функција $f : F_2^n \to F_2$ може се јединствено представити као полином у пољу $F_2[x_1, x_2, \ldots, x_n]/(x_1^2 = x_1, \ldots, x_n^2 = x_n)$ у облику

$$f(x) = \sum_{a \in F_2^n} c_a * x_1^{a_1} * x_2^{a_2} * \ldots * x_n^{a_n}$$

где су $x_1, x_2, \ldots, x_n$ променљиве, коефицјенти c_a су из F_2, $a = (a_1, a_2, \ldots, a_n)$ је вектор из F_2^n и $x_i^{a_i} = x_i$ за $a_i = 1$.

За довољно мало n коефицјенти полинома могу се одредити директним приступом. Пример израчунавања коефицјената АНФ илустрован је табелом 2.2, а пример АНФ једне функције илустрован је табелом 2.3

Табела 2.3: Пример АНФ за функцију арности 3

$x \in F_2^n$	$f(x)$	ANF
000	0	0
001	1	1
010	0	0
011	1	0
100	0	0
101	1	0
110	1	1
111	0	0

За веће вредности n директни приступ није практично применљив. Стога се АНФ најчешће добија из истинитосних таблица применом везе $ANF_f = A_n * f \mod 2$ где је A_n матрица димензија 2^n x 2^n која се дефинише рекурзивно изразом

$$A_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes A_{n-1} \qquad A_0 = |1|$$

У наставку су дата нека важна алгебарска својства Булових функција везана за АНФ. На основу ових својстава лако се могу дефинисати различити скупови Булових функција. Она су неизоставна приликом било које класификације Булових функција.

Дефиниција 12 *Алгебарска тежина функције f, у ознаци $awt(f)$ представља број чланова полинома ANF_f.*

Дефиниција 13 *Алгебарски степен функције f, у ознаци $ord(f)$ представља степен полинома ANF_f.*

Дефиниција 14 *Булова функција је хомогена ако су сви чланови њене алгебарске нормалне форме истог степена.*

2.2.3 Рид Малерови кодови

Некада је потребно генерисати све Булове функције одређеног степена. Чување великог броја репрезентација није ефикасно. Због тога је погодно користити Рид Малерове (*Reed–Muller*) кодове као алат за генерисање скупа Булових функција задатог степена. Рид Малерови кодови припадају групи кодова за корекцију грешака и њихова примена је много шира од онога што је представљено у овом раду. За детаљан опис погледати [3].

Табела 2.4: Пример матрице генератора за $R(3,3)$

1	1	1	1	1	1	1	1	1
x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
$x_1 * x_2$	0	0	0	1	0	0	0	1
$x_1 * x_3$	0	0	0	0	0	1	0	1
$x_2 * x_3$	0	0	0	0	0	0	1	1
$x_1 * x_2 * x_3$	0	0	0	0	0	0	0	1

Дефиниција 15 *Рид Малеров код дужине n и реда k представљен је скупом $R(k,n) = \{f \in p_n \ : ord(f) \leq k\}$.*

Рид Малеров код дужине n може се генерисати из матрице генератора. Димензија матрице генератора је

$$(\sum_{k=\overline{1,n}} \binom{n}{k} + 1) \times n$$

Прва врста матрице одговара јединичној функцији. Наредних n врста редом представља улазне променљиве $x_1, x_2, \ldots, x_n$. Остале врсте представљају редом производе од по $2, 3, \ldots, n$ чланова, где су чланови производа неке од првих n врста. Пример матрице генератора за $R(3,3)$ представљен је табелом 2.4.

Теорема 16 *Скуп свих линеарних комбинација врста матрице генератора једнак је скупу $R(n,n)$*

Доказ. (ППС) Нека постоји елемент $f(x) \in R(n,n)$ који није линеарна комбинација врста матриве генератора. Тада полином ANF_f мора садржати бар један члан степена већег од n јер су сви чланови степена мањег од n садржани у матрици генератору, али тада не би важило $f(x) \in R(n,n)$. ■

Рид Малерови кодови представљају једну класификацију Булових функција. На основу њих могуће је дефинисани многе класе Булових функција. За даљи рад интересантни су скупови косета Рид Малеровог кода задатог реда.

Дефиниција 17 *За целе бројеве* r, s *и* n *такве да је* $0 \leq s < r \leq n$ *дефинише се скуп косета* $R(r,n)$ *по* $R(s,n)$ *као* $R(r,n)/R(s,n) = \{R(r,n) * f(x) | \forall f(x) \in R(s,n)\}$.

2.3 Афино еквивалентне Булове функције и инваријанте

У овој тачки даје се преглед основних појмова везаних за афина пресликавања и еквивалентност Булових функција. Детаљно се разматрају трансформације афино еквивалентних Булових функција и везе међу резултатима трансформација. Акценат је стављен на поређење инваријанти скупа Булових функција и скупа функција добијених применом трансформација.

Прво треба дефинисати афина пресликавања и афину еквивалентност, а затим се може дефинисати и инваријанте скупа Булових функција. Дејство елемента $a = (A, b) \in AGL(n, 2)$ на Булове функције из p_n дефинише се изразом:

$a : p_n \to p_n$
$f(x) \to f(x * A + b)$

Дефиниција 18 *Булове функције* $f(x), g(x) \in p_n$ *су афино еквивалентне ако постоји* $(A, b) \in AGL(n, 2), l \in F_2^n, \rho \in F_2$ *тако да важи* $g(x) = f(xA + b) + l \bullet x + c$.

Дефиниција 19 *Инваријанта скупа* p_n *је свако пресликавање* M *из* p_n *у неки скуп, такво да за сваке две еквивалентне функције* $f(x), g(x) \in p_n$ *важи* $M(f) = M(g)$.

2.3.1 Волшова трансформација

Волшова трансформација (*Joseph L. Walsh, Hans Rademacher and Jacques Hadamard*) пружа још један начин за јединствено представљање функција. Резултат примене Волшове трансформације на функцију f назива се Волшов спектар функције f. Волшова тренсформација ради са целобројном векторима, па се као специјалан случај може применити и на Булове функције. На крају се показује да се скуп апсолутних вредности Волшовог спектра може искористити као инваријанта.

Дефиниција 20 *Волшова трансформација функције f, у ознаци W_f, добија се из истинитосне таблице функције f применом израза*

$$W_f(w) = \sum_{x \in F_2^n} (-1)^{f(x)} * (-1)^{w \bullet x}$$

.

Ако је функција задата поларном истинитосном таблицом израз за Волшову трансформацију може се записати двојако.

$$W_f(w) = \sum_{x \in F_2^n} (-1)^{f(x)} * (-1)^{w \bullet x}$$

или

$$W_f(w) = \sum_{x \in F_2^n} f(x) * (-1)(-1)^{w \bullet x}$$

јер је $(-1)^{\widehat{f(x)}} = \widehat{f}(x)$.

Из дефиниције следи да Волшов спектар функције узима вредности $-2^n \leq W_f(w) \leq 2^n$ за свако $w \in F_2^n$. Израз за Волшову трансформацију може се записати у матричном облику $W_f(w) = M_n * f(w)$, где је $_n$ Волшова матрица реда 2^n дефинисана рекурзивно изразом

$$M_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes M_{n-1} \qquad M_0 = |1|$$

Инверз матрице M_n је матрица $\frac{1}{2^n} * M_n$. Пример Волшовог спектра илустрован је табелом 2.5.

Волшова трансформација је инвертибилна у оба поменута случаја. Непосредно се проверава $\frac{1}{2^n} * M_n * W_f(w) = f(w)$.

Израчунавање Волшовог спектра по претходним формулама није ефикасно за функције велике арности. Израчунавање захтева 2^{2^n} операција. Бржи начин израчунавања је брза Волшова трансформација, која се заснива на идеји растављања Волшове матрице у облик $M_n = A_n^1 * A_n^2 * A_n^3 * \ldots * A_n^n$, где је $A_n^i = I_{2^{n-i}} \otimes M_1 \otimes I_{2^{i-1}}$, где је I_k јединична матрица реда k. Брза Волшова трансформација захтева $n2^n$ операција. Унаставку је дат доказ теореме о разлагању Волшове матрице реда n.

Теорема 21 $M_n = A_n^1 * A_n^2 * A_n^3 * \ldots * A_n^n$, *где је* $A_n^i = I_{2^{n-i}} \otimes M_1 \otimes I_{2^{i-1}}$

Табела 2.5: Пример Волшовог спектра за функцију арности 3

$x \in F_2^n$	$f(x)$	$\overline{f}(x)$	$W_f(x)$
000	0	1	0
001	1	−1	4
010	0	1	0
011	1	−1	4
100	0	1	0
101	1	−1	4
110	1	−1	0
111	0	1	−4

Доказ. Индукцијом по n.
База индукције $n = 1$

$$M_1 = A_1^1 = (I_1 \otimes M_1 \otimes I_1) = [1] \otimes [1] \otimes [1] = [1]$$

Нека тврђење важи за n. Тада за $1 \leq i \leq n$ важи

$$\begin{aligned} A_{n+1}^i &= I_{2^{n+1-i}} \otimes M_2 \otimes I_{2^{i-1}} \\ &= I_2 \otimes I_{2^{n-i}} \otimes M_2 \otimes I_{2^{i-1}} \\ &= I_2 \otimes A_n^i \end{aligned}$$

$$A_{n+1}^{n+1} = M_2 \otimes I_{2^m}$$

Тада је

$$\begin{aligned} A_{n+1}^1 * \ldots * A_{n+1}^{n+1} &= (I_2 \otimes A_n^1) * \ldots * (I_2 \otimes A_n^n) * (M_2 \otimes I_{2^m}) \\ &= M_2 \otimes (A_n^1 * \ldots * A_n^n) \\ &= M_2 \otimes M_2^n \\ &= M_2^{n+1} \end{aligned}$$

■

Преко Волшовог спектра лако се дефинишу многа криптографска својства Булових функција. У наставку је дато неколико теорема и дефиниција које илуструју моћ овог једноставног алата.

Теорема 22 *Ако је $W_f(0) = 0$ функција је утавнотежена.*

Доказ. Заменом $w = 0$ у изразу 20 добија се:

$$\begin{aligned} W_f(0) &= \sum_{x \in F_2^n} (-1)^{f(x)} * (-1)^{0 \bullet x} \\ &= \sum_{x \in F_2^n} (-1)^{f(x)} * (-1)^{0 \bullet x} \\ &= \sum_{x \in F_2^n} (-1)^{f(x) \oplus 0} \\ &= \sum_{x \in F_2^n} (-1)^{f(x)} \end{aligned}$$

Вредност овог израза биће 0 само ако је функција уравнотежена. ■

Дефиниција 23 *Максимална апсолутна вредност W_f представља меру нелинеарности Булове функције. Нелинеарност представља удаљеност функције од најближе афине функције.*

Дефиниција 24 *Булова функција је корелационо имуна реда k,односно $f \in CI(k)$ ако за њен Волшов спектар важи $W_f(w) = 0$ за $1 \leq wt(w) \leq k$.*

Теорема 25 *Скуп апсолутних вредност Волшовог спектра је инваријанта скупа p_n.*

Доказ. Видети доказ става 26. ■

Овде ће се разматрати Волшова трансформација афино еквивалентних Булових функција. Показује се да је скуп апсолутних вредности Волшовог спектра инваријанта скупа p_n.

Став 26 *Нека су n, r цели бројеви такви да је $1 \leq r \leq n$. Нека су дате афино еквивалентне функције $f(x)$ и $g(x) = f(xA + b) \oplus l \bullet x \oplus c$, где је $c \in F_2, x, b, l \in F_2^n, A \in GL(n, 2)$. За све $w \in F_2^n$ важи*

$$W_g(w) = (-1)^{(b * A^{-1}) \bullet (l+w) \oplus c} * W_f((l + w) * A^{-1T})$$

.

Доказ.

$$W_g(w) = \sum_{x \in F_2^n} (-1)^{g(x)} * (-1)^{w \bullet x}$$

Увођењем смене $x = (y + b)A^{-1}$ добија се:

$$\begin{aligned}
W_g(w) &= \sum_{y \in F_2^n} (-1)^{f(y) \oplus l \bullet ((y+b)A^{-1}) \oplus c} * (-1)^{w \bullet ((y+b)A^{-1})} \\
&= \sum_{y \in F_2^n} (-1)^{f(y) \oplus l \bullet ((y+b)A^{-1}) \oplus c \oplus w \bullet ((y+b)A^{-1})} \\
&= \sum_{y \in F_2^n} (-1)^{f(y) \oplus (l+w) \bullet ((y+b) * A^{-1}) \oplus c} \\
&= \sum_{y \in F_2^n} (-1)^{f(y) \oplus (l+w) \bullet (y * A^{-1} + b * A^{-1}) \oplus c} \\
&= \sum_{y \in F_2^n} (-1)^{f(y) \oplus (y * A^{-1}) \bullet (l+w) \oplus (b * A^{-1}) \bullet (l+w) \oplus c} \\
&= (-1)^{(b * A^{-1}) \bullet (l+w) \oplus c} * \sum_{y \in F_2^n} (-1)^{f(y) \oplus y \bullet ((l+w) * A^{-1T})} \\
&= (-1)^{(b * A^{-1}) \bullet (l+w) \oplus c} * W_f((l + w) * A^{-1T})
\end{aligned}$$

■

Последица 27 *Апсолутна вредност Волшовог спектра функције* $f(x)$ *на позицији* i *једнака је апсолутној вредности Волшовог спектра функције* $g(x)$ *на позицији* j *за* $i = jA^T + l$. *Расподела скупа апсолутних вредности Волшовог спектра је инваријанта скупа* p_n.

2.3.2 Аутокорелација Булове функције

Друга инваријанта скупа Булових функција заснива се на њиховој аутокорелацији. Аутокорелација Булове функције представљена је функцијом аутокорелације. Функција аутокорелације је специјалан случај функције корелације, када се тражи корелација функције са самом собом. Скуп вредности функција аутокорелације Булове функције f назива се аутокорелациони спектар функције f.

Дефиниција 28 *Функција корелације две Булове функције* $f, g \in p_n$ *дефинисана је изразом*

$$c_{fg}(w) = \sum_{x \in F_2^n} (-1)^{f(x)} * (-1)^{g(w+x)}, \qquad \forall w \in F_2^n$$

.

Табела 2.6: Пример функције аутокорелације за функцију f арности 3

$x \in F_2^n$	$f(x)$	$\overline{f}(x)$	$c_f(x)$
000	0	1	8
001	1	−1	−8
010	0	1	0
011	1	−1	0
100	0	1	0
101	1	−1	0
110	1	−1	0
111	0	1	0

Дефиниција 29 *Аутокорелација Булове функције $f \in p_n$ дефинисана је изразом*

$$c_f(s) = \sum_{x \in F_2^n} (-1)^{f(x)} * (-1)^{f(x+s)}, \qquad \forall s \in F_2^n$$

Пример функције аутокорелације илустрован је табелом 2.6.

Функција аутокорелације повезана је са Волшовом трансформацијом *Wiener–Hinchin*-овом теоремом. Ова теорема је од практичног значаја, јер пружа могућност ефикасне програмске реализације функције аутокорелације.

Теорема 30 *(Wiener–Hinchin) Свака Булова функција задовољава једнакост $W_{c_f}(w) = W_f^2(w) \qquad \forall w \in F_2^n$, где је квадрирање операција из скупа целих бројева.*

Доказ. Ради лакшег записа нека је функција f дата поларном истинитосном таблицом.

$$\begin{aligned} W_{c_f}(w) &= \sum_{s \in F_2^n} c_f(s) * (-1)^{w \bullet s} \\ &= \sum_{s \in F_2^n} \sum_{x \in F_2^n} (-1)^{f(x) \oplus f(x+s)} * (-1)^{w \bullet s} \\ &= \sum_{x \in F_2^n} \sum_{s \in F_2^n} (-1)^{f(x) \oplus f(x+s) \oplus w \bullet s} \end{aligned}$$

Како је скуп свих вектора из F_2^n инваријанта за сваку примењену трансформацију, може се узети $s = x \oplus s$ па се добија:

$$\begin{aligned} W_{c_f}(w) &= \sum_{x \in F_2^n} \sum_{s \in F_2^n} (-1)^{f(x) \oplus f(s) \oplus w \bullet (x+s)} \\ &= \sum_{x \in F_2^n} (-1)^{f(x) \oplus w \bullet x} * \sum_{s \in F_2^n} (-1)^{f(s) \oplus w \bullet s)} \\ &= \sum_{x \in F_2^n} f(x) * (-1)^{w \bullet x} * \sum_{s \in F_2^n} f(s) * (-1)^{w \bullet s)} \\ &= W_f(w) * W_f(w) = W_f^2(w) \end{aligned}$$

■

Овде ће се разматрати функција аутокорелације афино еквивалентних Булових функција. Испоставља се да се скуп апсолутних вредности функције аутокорелације може послужити као инваријанта.

Став 31 *Нека су n, r цели бројеви такви да је $1 \leq r \leq n$ и нека су дате афино еквивалентне функције $f(x)$ и $g(x) = f(x*A+b) \oplus l \bullet x \oplus c$, где је $c \in F_2, x, b, l \in F_2^n, A \in GL(n,2)$. За све $s \in F_2^n$ важи $c_g(s) = (-1)^{l \bullet s} c_f(s * A)$*

Доказ.

$$\begin{aligned} c_g(s) &= \sum_{x \in F_2^n} (-1)^{f(x*A+b) \oplus l \bullet x \oplus c} * (-1)^{f((x+s)A+b) \oplus l \bullet (x+s) \oplus c} \\ &= \sum_{x \in F_2^n} (-1)^{f(x*A+b) \oplus l \bullet x \oplus c \oplus f((x+s)A+b) \oplus l \bullet x \oplus l \bullet s \oplus c} \\ &= \sum_{x \in F_2^n} (-1)^{f(x*A+b) \ominus f((x+s)A+b) \oplus l \bullet s} \\ &= \sum_{x \in F_2^n} (-1)^{f(x*A+b) \ominus f(x*A+s*A+b)} * (-1)^{l \bullet s} \\ &= \sum_{x \in F_2^n} (-1)^{l \bullet s} * c_f(s * A) \end{aligned}$$

■

Последица 32 *Апсолутна вредност аутокорелационог спектра функције $f(x)$ на позицији i једнака је апсолутној вредности аутокорелационог спектра функције $g(x)$ на месту j за $i = jA + l$. Расподела скупа апсолутних вредности аутокорелационог спектра је инваријанта скупа p_n.*

2.3.3 Извод Булове функције

Извод представља осетљивост функције на промене вредности задатог скупа променљивих. Извод Булове функције по задатом вектору одговара парцијалном изводу функције по променљивима за које задати вектор има вредност 1. На овај начин се прелази на простор мањих диманзија. Испоставља се да је понашање афино еквивалентних функција исто за изводе по задатом вектору и његовој афиној слици.

Дефиниција 33 *Извод Булове функције f по вектору $u \in F_2^n$ је Булова функција $D_u(f)$ дефинисана изразом $D_u(f) = f(x+u) \oplus f(x)$.*

Теорема 34 *[4] Извод Булове функције има следећа својства:*

1. $D_u(f+g) = D_u(f) \oplus D_u(g)$
2. $D_u(f \circ B) = D_{A*u}(f) \circ B$ *ако је* $B \in AGL(n,2)$
3. $D_{u+v}(f) = D_u(f) \oplus D_v(f) \oplus D_{u,v}(f)$

Доказ. Према дефиницији 33

1.

$$\begin{aligned} D_u(f+g) =& f(x) \oplus g(x) \oplus f(x+u) \oplus g(x+u) \\ =& f(x) \oplus f(x+u) \oplus g(x) \oplus g(x+u) \\ =& D_u(f) \oplus D_u(g) \end{aligned}$$

2.

$$\begin{aligned} D_u(f \circ B) =& f \circ B(x) \oplus f \circ B(x+u) \\ =& f(x*A+b) \oplus f((x+u)*A+b) \\ =& f(x*A+b) \oplus f(x*A+u*A+b) \\ =& D_{u*A}(f)(x*A+b) \\ =& D_{u*A}(f) \circ B \end{aligned}$$

3.

$$\begin{aligned}D_u(f) \oplus D_v(f) \oplus D_{u,v} &= f(x) \oplus f(x+u) \oplus f(x) \oplus f(x+v) \\ &\quad \oplus D_v(f(x) \oplus f(x+u)) \\ &= f(x+u) \oplus f(x+v) \oplus D_v(f(x)) \oplus D_v(f(x+u)) \\ &= f(x+u) \oplus f(x+v) \oplus f(x) \oplus f(x+v) \oplus f(x+u) \\ &\quad \oplus f(x+u+v) \\ &= f(x) \oplus f(x+u+v) \\ &= D_{u+v}(f)\end{aligned}$$

■

Последица 35 *Извод Булове функције $f(x)$ у тачки j једнак је изводу Булове функције $g(x) = f(x) \circ B, B = (A, b) \in AGL(n, 2)$ у тачки i када је $j = iA$.*

На основу следећег става показано је да инваријанта скупа пролази кроз извод функције.

Став 36 *Нека је са δ_v означено пресликавање из скупа $R(r,n)/R(s,n)$ у скуп $R(r-1,n)/R(s-1,n)$ дефинисано изразом $f \to D_v(f)$ и нека је M_1 инваријанта скупа $R(r-1,n)/R(s-1,n)$. Расподела скупа вредности $M_1 \circ \delta_v$ је инваријанта скупа $R(r,n)/R(s,n)$.*

Доказ. Нека је M_1 инваријанта $R(r,n)/R(s,n)$ и нека $A \in GL(n,2)$. За сваку функцију $f \in R(r,n)/R(s,n)$ на основу теореме 34 добија се $M_1(D_u(f \circ A)) = M_1(D_{uA}(f) \circ A) = M_1(D_{uA}(f))$. Одавде се добија да су расподеле вредности $M_1(D_u(f))$ и $M_1(D_u(f \circ A))$исте. ■

Последица 37 *Инваријанта скупа $R(r,n)/R(s,n)$ преноси се у скуп $R(r-1,n)/R(s-1,n)$. Слично се може дефинисати за скупове $R(r-1,n)/R(s-1,n)$ и $R(r-2,n)/R(s-2,n)$ итд. Инваријанта се преноси у скуп са мањим бројем променљивих.*

2.3.4 Разлагање Булове функције

Још један начин за прелазак у простор мањих димензија заснива се на разлагању функција по задатом скупу променљивих. Ако се функција разлаже по једној променљивој за сваку њену могућу вредност добија се по једна функција за 1 мање арности од полазне функције. У случају Булових функција добијаће се две нове функције. Разлагање Булове функције према задатом вектору представља скуп разлагања функције по свакој променљивој која у задатом вектору има вредност 1.

Лема 38 *[6] Нека је дата Булова функција $f(x) \in p_n$. Разлагање функције по првој компоненти дефинисано је као*

$$f(x) = (x_1 \oplus 1) * f_0(x') \oplus x_1 * f_1(x')$$

где је $x' = (x_2, x_3 \dots x_n)$ и $f_0, f_1 : F_2 \to F_2^{n-1}$.

У наставку је показано да се две афино еквивалентне функције по задатом модулу могу разложити тако да су функције добијене разлагањем такође афино еквивалентне по задатом модулу.

Теорема 39 *Нека су n, r цели бројеви такви да је $1 \leq r \leq n$. Нека су дате афино еквивалентне функције $f(x)$ и*

$$g(x) = f(x * A + b) \mod R(s, n), x, b \in F_2^n, A \in GL(n, 2)$$

Тада се функција $g(x)$ може записати у облику

$$g(x) = (x * C_1 \oplus b_1 \oplus 1) * f_0(x'') \oplus (x * C_1 \oplus b_1) * f_1(x'')$$

где је $x'' = (x*C_2 \oplus b_2, x*C_3 \oplus b_3, \dots, x*C_n \oplus b_n)$ и $C_1, C_2, \dots, C_n$ су колоне матрице A. Тада су функције $f_0(x', f_1(x')$ и $f_0(x''), f_1(x'')$ еквивалентне по модулу $R(s, n-1)$.

Доказ. $f(y) = (y_1 \oplus 1) * f_0(y') \oplus y_1 * f_1(y'), y' = (y_2, y_3 \dots y_n)$
Увођењем смене $y = x * A + b$ добија се
$f(x * A + b) = (x * C_1 \oplus b_1 \oplus 1) * f_0(x * C_2 \oplus b_2, x * C_3 \oplus b_3, \dots, x * C_n \oplus b_n) \oplus (x * C_1 \oplus C_1 \oplus 1) * f_1(x * C_2 \oplus b_2, x * C_3 \oplus b_3, \dots, x * C_n \oplus b_n)$, где су $C_1, C_2, \dots, C_n$ колоне матрице A
$g(x) = (x * C_1 \oplus b_1 \oplus 1) * f_0(x * C_2 \oplus b_2, x * C_3 \oplus b_3, \dots, x * C_n \oplus b_n) \oplus (x * C_1 \oplus C_1 \oplus 1) * f_1(x * C_2 \oplus b_2, x * C_3 \oplus b_3, \dots, x * C_n \oplus b_n)$. ■

У наставку је показано да афина еквивалентност функција по задатом модулу пролази кроз разлагање функција по задатом вектору и његовој афиној слици.

Став 40 *Нека су дате афино еквивалентне функције* $f(x)$ *и* $g(x) = f(x*A+b) \mod R(s,n), x, b \in F_2^n, A \in GL(n,2)$ *и нека је функција* $f(x)$ *растављена на две функције* $f_{a\bullet x=0}(x)$ *и* $f_{a\bullet x=1}(x)$ *по вектору* a. *Функција* $g(x)$ *може се раставити по вектору* c *на две функције* $g_{c\bullet x=0}(x)$ *и* $g_{c\bullet x=1}(x)$ *тако да су* $g_{c\bullet x=0}(x)$ *и* $g_{c\bullet x=1}(x)$ *еквивалентне функцијама* $f_{a\bullet x=0}(x)$ *и* $f_{a\bullet x=1}(x)$ *по модулу* $R(s,n-1)$ *када је* $c = a * A^T$.

Доказ. Према леми 38 раставља се функција $f(x)$ на функције $f_{a\bullet x=0}(x)$ и $f_{a\bullet x=1}(x)$. Према теореми 39 функција $g(x)$ раставља се по вектору $a \bullet (x * A + b) = 0$ и $a \bullet (xA + b) = 1$ на две функције $g_{c\bullet x=0}(x)$ и $g_{c\bullet x=1}(x)$ еквивалентне функцијама $f_{a\bullet x=0}(x)$ и $f_{a\bullet x=1}(x)$ по модулу $R(s,n-1)$. Како је $a \bullet (x * A + b) = x \bullet (a * A^T) + a \bullet b$ добија се да је $c = a * A^T$. ■

Остало је још да се покаже да се инваријанта преноси у простор мањих димензија ако се као средство преласка користи разлагање.

Став 41 *Нека је* δ_v *пресликавање из* $R(r,n)/R(s,n)$ *у* $R(r-1,n)/R(s-1,n)$ *дефинисано изразом* $f \to (f_{v\bullet x=0}, f_{v\bullet x=1})$ *и нека је* M_1 *инваријанта скупа* $R(r-1,n)/R(s-1,n)$. *Расподела скупа вредности* $M_1 \circ \delta_v$ *је инваријанта скупа* $R(r,n)/R(s,n)$.

Доказ. Нека је M_1 инваријанта $R(r,n)/R(s,n)$ и нека $A \in GL(n,2)$. За сваку функцију $f \in R(r,n)/R(s,n)$ на основу теореме 39 добија се

$$M_1((f_{v\bullet x=0}, f_{v\bullet x=1})(f \circ A^{-1})) = M_1((f_{v\bullet A^{-1}*x=0}, f_{v\bullet A^{-1}*x=1})(f))$$

одакле се добија да су расподеле вредности $M_1(f_{v\bullet x=0}, f_{v\bullet x=1})$ и $M_1(f_{v\bullet A^{-1}*x=0}, f_{v\bullet A^{-1}*x=1})$ једнаке. ■

2.3.5 Скуп суседних функција

Булова функција може бити јединствено представљена скупом суседних функција. У наставку су упоређени скупови суседних функција две афино еквивалентне функције.

Дефиниција 42 *[6] За $i \in F_2^n$ суседна функција функције $f(x)$ дефинише се изразом* $f_i(x) = \begin{cases} f(x) & , x \neq i \\ f(x)+1 & , x = i \end{cases} i \in F_2^n$

Овде је показано да су функције суседства афино еквивалентних функција такође афино еквивалентне до на пермутацију.

Став 43 *Нека су дате афино еквивалентне Булове функције $f(x)$ и $g(x) = f(x * A + b) \oplus l \bullet x \oplus c, A \in GL(2,n), x, b, l \in F_2^n, c \in F_2$, и нека је $i, j \in F_2^n$. За њихове суседне функције важи једнакост $g_j(x) = f_i(x * A + b) \oplus l \bullet x \oplus c$ када је $j * A = i + b$.*

Доказ.

$$f_i(x * A + b) \oplus l \bullet x \oplus c = \begin{cases} f_i(x * A + b) \oplus l \bullet x \oplus c & , x * A + b \neq i \\ f_i(x * A + b) \oplus l \bullet x \oplus c \oplus 1 & , x * A + b = i \end{cases} \quad i \in F_2^n$$

$$= \begin{cases} g_i(x) & , x \neq (i+b) * A^{-1} \\ g(x) \oplus 1 & , x = (i+b) * A^{-1} \end{cases} \quad i \in F_2^n$$

Одавде следи да је $g_j(x)$ еквивалентна $f_i(x * A + b) \oplus l \bullet x \oplus c$ када је $j = (i+b) * A^{-1}$ за $i, j \in F_s^n$. ■

Став 44 *Нека је дата Булова функција $f(x) \in R(r,n)$ и нека је M_1 инваријанта скупа $R(n,n)/R(1,n)$. Тада је скуп $\{M_1(f_i(x)) | i \in F_2^n\}$ инваријанта скупа $R(r,n)/R(1,n)$.*

Глава 3

Алгоритам за утврђивање афине еквивалентности двеју датих Булових функција

Ова глава подељена је у два дела. У првом делу приказује се алгоритам из рада [1] за утврђивање афине еквивалентности двеју задатих Булових функција. У другом делу главе дати су анализа алгоритма и неке смернице за програмску реализацију.

3.1 Алгоритам

Алгоритми за проверу афине еквивалентности функција базирају се на умањењу скупа кандидата за проверу. Овај алгоритам као алат за умањење скупа кандидата користи инваријанте и трансформације Булових функција описане у глави 2. Алгоритам на улазу прихвата две булове функције, а на излазу даје све тројке A, b, l ако су функције афино еквивалентне. Уколико функције нису афино еквивалентне излаз је порука о томе. У наставку је дат псеудо код алгоритма.

Улаз: две Булове функције $f(x)$ и $g(x)$ арности n
Излаз: $A \in Gl(2, n), b, l \in F_2^n$ ако су Булове функције $f(x)$ и $g(x)$ афино еквивалентне, у супротном порука да функције нису афино еквивалентне

1. Израчунати Волшов и аутокорелациони спектар функција f и g. Упоредити расподеле апсолутних вредности Волшовог и аутокорелационог спектра функција f и g. Уколико расподеле нису једнаке издати поруку да функције нису афино еквивалентне.

2. Нека су са e_i обележени јединични вектори из F_2^n. За све вредности $c_{e_i}(g)$ аутокорелационог спектра функције g проћи по свим елементима $v \in F_2^n$ и када год важи $|c_{e_i}(g)| = |c_v(f)|$ сачувати вектор v као кандидат за $n-i$-ту врсту матрице A.

3. Нека су са e_i обележени јединични вектори из F_2^n. За сва разлагања функције f по јединичним векторима e_i на две функције $f_{e_i \bullet x=0}$ и $f_{e_i \bullet x=1}$ проћи по свим елементима $v \in F_2^n$ и разложити функцију g на две функције $g_{v \bullet x=0}$ и $g_{v \bullet x=1}$ по вектору v. Када год је

$$|WT(f_{e_i \bullet x=0}, f_{e_i \bullet x=1})| = |WT(g_{v \bullet x=0}, g_{v \bullet x=1})|$$

сачувати вредност v као кандидат за i-ту колону матрице A.

4. За све јединичне векторе израчунати извод функције g за вектор e_i. За сваки израчунат извод функције g проћи по свим елементима $v \in F_2^n$ и израчунати извод функције f за вектор v. Када год је $|WT(d_(e_i)(g))| = |WT(d_(v)(f))|$ сачувати вектор v као кандидат за $n-i$-ту врсту матрице A.

5. Користећи скупове кандидата за врсте и колоне матрице A формирати скуп кандидата регуларних матрица.

6. За сваки јединични вектор израчунати суседну функцију $lc_{e_i}(g)$ функције g. За сваку израчунату суседну функције проћи по свим елементима $v \in F_2^n$ и израчунати суседну функцију $lc_v(f)$ функције f. Када год је $|WT(lc_{e_i}(g))| = |WT(lc_v(f))|$ треба проћи по свим елементима скупа кандидата матрица и за сваку матрицу сачувати кандидат за вектор b у облику $e_i * A + v$.

7. За све јединичне векторе узети вредност Волшовог спектра $W_f(e_i)$ функције f и за сваку узету вредност проћи по свим елементима $v \in F_2^n$. Када год је $|W_f(e_i)| = |W_g(v)|$ за сваку матрицу кандидата треба сачувати кандидат за вектор l у облику $e_i * A + v$.

8. Проћи по свим елементима скупа кандидата матрица A и њима договарајућих скупова кандидата за векторе b и l и проверити да ли важи $g(x) = f(x*A+b) \oplus l \bullet x$. Уколико једнакост важи, сачувати тројку A, b, l у скуп резултата. По завршетку свих провера вратити скуп резултата.

3.2 Анализа алгоритма

У овој тачки дата је анализа представљеног алгоритма са акцентом на ефикасност алгоритма. Анализом најгорег случаја добија се да је сложеност алгоритма експоненцијална реда $O(n*2^n)$ за функције арности n. У најгорем случају алгоритам ради као алгоритам грубе силе покушавајући са свим могучим комбинацијама параметара A, b и l.

Ефикасност алгоритма зависи од коришћених инваријанти. Инваријанте се могу одабрати тако да кардиналност свих скупова кандидата не буду 2^n или $2^(n-1)$. Уколико су кардиналности скупова једнаке 2^n или $2^(n-1)$ ефикасност алгоритма једнака је ефикасности алгоритма грубе. Ипак, вероватноћа да све кардиналности имају вредност 2^n или $2^(n-1)$ је јако мала. Због тога је ефикасност овог алгоритма скоро увек боља од ефикасности алгоритма грубе силе. Други начин да са побољша ефикасност је избор других n линеарно независних вектора уместо узетих јединичних вектора. За ова побољшања потребно је детаљно анализирати различите класе Булових функција и пронаћи оне векторе и инваријанте који за одређену класу дају најмањи број кандидата. Овде је остављен простор за унапређење алгоритма.

Приликом програмске реализације већина трансформација Булових функција можеу се реализовати рекурзивно. Волшова трансформација може се реализовати као брза Волшова трансформација из које се може добити и функција аутокорелације користећи *Wiener–Hinchin*-ову теорему. За све инваријанте може се искористити расподела скупа апсолутних вредности Волшовог спектра. То неће увек дати добре резултате. На пример, ако су на улазу дате бент функције за које важи да је вредност Волшовог спектра увек иста, алгоритам неће успети да умањи скупове кандидата.

Глава 4

Програмска реализација

Ова глава подељена је на два дела. У првом делу дат је опис програмске реализације алгоритма програмском језику *C++*. Други део главе покрива програмску реализацију и резултате тестирања алгоритма.

4.1 Програмска реализација алгоритма

Пре саме реализације требало је одабрати инваријанту и скуп јединичних вектора из F_2^n који ће се искористити за генерисање скупа кандидата тројки A, b, l. Као основна инваријанта одабран је скуп апсолутних вредности Волшовог спектра функције. За линеарно независне векторе одабрна је скуп јединичних вектора из F_2^n. За реализацију алгоритма одабран је програмски језик *C++* због многобројних класа погодних за рад са низовним типовима података.

Срж реализације чини класа којом су представљене Булове функције. Реализована је и помоћна класа за представљање матрице са методима погодним за рад са врстама и колонама. Булова функција представљена је класом *BooleanFunction*. Атрибути објекта класе *BooleanFunction* су вектор битова који представља истинитосну таблицу Булове функције и два неозначена цела броја који представљају арност функције и број елемената истинитосне таблице. Све трансформације Булове функције које се користе у алгоритму реализоване су као јавне методе класе *BooleanFunction*. За детаљан опис класе *BooleanFunction* погледати датотеку *BooleanFunction.h* дату у прилогу.

Реализација метода подељена је у три дела. Први део чине методи који реализују трансформације Булових функција. У другом делу дат

је преглед метода за генерисање кандидата тројки A, b, l. Трећи део представља реализацију метода који даје одговор да ли су Булове функције афино еквивалентне или не.

4.1.1 Методи за реализацију трансформација Булових функција

Сви методи за реализацију трансформација Булових функција позивају се над објектом класе *BooleanFunction*. У наставку су дати прототипови и описи метода. Детаљан опис метода налази се у датотеци *BooleanFunction.cpp* датој у прилогу.

Волшова трансформација

Прототип метода који реализује Волшову трансформацију:

```
std::vector<int> BooleanFunction::fastWalshTransform();
```

Волшова трансформација реализована је као брза Волшова трансформација. Метода се позива над објектом класе *BooleanFunction* и враћа објекат класе *std::vector<int>* који представља Волшов спектар Булове функције. Метода је написана тако да се сва израчунавања обављају у месту. У методу се најпре постављају промењиве за одређивање позиција вектора које ће мењати вредност, а потом се позива потпрограм који врши потребне операције над вредностима. Реализована је и метода за израчунавање инверзне Волшове трансформације. Њу није потребно посебно наводити, јер се од методе за добијање Волшове трансформације разликује само на крају, када се сваки члан вектора који функција враћа дели са вредношћу 2^n.

Функција аутокорелације

Прототип метода који реализује функцију аутокорелације:

```
std::vector<int> BooleanFunction::autocorrelationFunction();
```

Метода се позива над објектом класе *BooleanFunction* и враћа објекат класе *std::vector<int>* који садржи аутокорелациони спектар функције. Ова метода реализована је помоћу методе за добијање Волшове трансформације. Искоришћена је *Wiener–Hinchin*-ова теорема. Прво се израчунава Волшов спектар функције. Затим се добијене вредности квадрирају, па се над њима позива метода за израчунавање инверзне Волшове трансформације. На тај начин се добијају вредности аутокорелационог спектра функције.

Разлагање Булове функције

Прототип метода који реализује разлагање Булове функције:

```
std::vector< std::pair<BooleanFunction, BooleanFunction> >
BooleanFunction::decomposeAtVectorV(unsigned v);
```

Метода се позива над објектом класе *BooleanFunction* и као аргумент узима један неозначен цео број који представља вектор по коме се функција разлаже. Метод враћа објекат типа *std::vector< std::pair<BooleanFunction, BooleanFunction> >* који садржи све парове функција на које је било могуће раставити полазну функцију по задатом вектору. На почетку се одређују промењиве које се могу заменити линеарном комбинацијом осталих промењивих, а затим се врши разлагање функције по прослеђеном вектору за сваку промењиву која може бити замењена. Вредност по којој се функција разлаже је скаларни производ вектора улаза функције и вектора прослеђеног као параметар функције.

Извод Булове функције

Прототип метода којим се реализује извод Булове функције по задатом вектору:

```
BooleanFunction BooleanFunction::getDerivateAtVectorV(unsigned v);
```

Метода се позива над објектом класе *BooleanFunction* и као аргумент узима један неозначен цео број који представља вектор по коме се тражи извод функције. Истинитосна таблица функције извода израчунава се у једном *for* циклусу. Метода враћа објекат типа *BooleanFunction* који представља функцију извода.

Суседне функције

Прототип метода за добијање суседне функције по задатом вектору:

```
BooleanFunction BooleanFunction::getFirstLocalConnection
FunctionAtVectorV(unsigned v);
```

Метода се позива над објектом класе *BooleanFunction* и као аргумент узима један неозначен цео број који представља вектор по коме се тражи суседна функција. Истинитосна таблица суседне функције израчунава се инвертовањем вредности истинитосне таблице полазне функције на масту прослеђеног параметра. Метода враћа објекат типа *BooleanFunction* који представља суседну функцију по задатом вектору.

4.1.2 Методи за генерисање кандидата

Сви методи за генерисање кандидата матрица и вектора реализовани су као статички методи. Прво ће бити наведени методи за генерисање матрица, а затим методи за генерисање вектора афиног пресликавања.

Генерисање кандидата за врсте матрице помоћу функције аутокорелације

Прототип метода за генерисање кандидата врста матрице помоћу функције аутокорелације:

```
std::vector< std::set<unsigned> > BooleanFunction::getCandidates
ForMatrixAFromAutocorrelation(std::vector<int> funF,
std::vector<int> funG, unsigned n);
```

Метода као аргументе прихвата два објекта типа *std::vector<int>* који представљају аутокорелационе спектре функција f и g и један неозначен цео број који представља дужину вектора спектра. Метода враћа објекат типа *std::vector< std::set<unsigned> >* који представља скупове кандидата за сваку врсту матрице A. Као инваријанта користи се рсподела скупа апсолутних вредност аутокорелационе функције. Кандидати се добијају поређењем апсолутних вредности аргумената на одређеним позицијама.

Генерисање кандидата за врсте матрице помоћу извода функције

Прототип метода за генерисање кандидата врста матрице помоћу извода Булове функције:

```
std::vector< std::set<unsigned> >BooleanFunction::getCandidates
ForMatrixAFromDervivation
(BooleanFunction f, BooleanFunction g);
```

Метода као аргументе прихвата два објекта типа *BooleanFunction* који представљају функције f и g, а враћа објекат типа *std::vector <std::set <unsigned> >* који представља скупове кандидата за сваку врсту матрице A. Као инваријанта користи се рсподела скупа апсолутних вредност Волшовог спектра. Током извршавања позива се метода за израчунавање извода Булове финкције за задати вектор. Пошто се израчунају потребни изводи функција и њихови Волшови спектри, извршиава се поређење расподела Волшових спектара извода и ако се расподеле поклопе, добија се кандидат једне врсте матрице A.

Генерисање кандидата за колоне матрице помоћу разлагања функције

Прототип метода за генерисање кандидата колона матрице помоћу разлагања функције:

```
std::vector< std::set<unsigned> > BooleanFunction::
getCandidatesForMatrixAFromDecomposition
(BooleanFunction f, BooleanFunction g);
```

Метода као аргументе прихвата два објекта типа *BooleanFunction* који представљају функције f и g, а враћа објекат типа *std::vector< std::set <unsigned> >* који представља скупове кандидата за сваку колону матрице A. Као инваријанта користи се рсподела скупа апсолутних вредност Волшовог спектра. Током извршавања позива се метода за разлагање функције по задатом вектору. Пошто се полазне функције разложе на парове функција, израчунавају се Волшови спектри парова и пореде се расподеле скупова њихових апсолутних вредности. Када год се расподеле поклопе добија се кандидат за једну од колона матрице A.

Генерисање скупа кандидата матрица

Прототип метода за генерисање скупа кандидата матрица:

```
std::vector<Matrix> BooleanFunction::candidatesA
(std::vector< std::set<unsigned> > rowCandidates,
 std::vector< std::set<unsigned> > colCandidates,
unsigned n);
```

Метода као аргументе прихвата два објекта типа *std::vector< std::set <unsigned> >* који представљају , а враћа објекат типа *std::vector< std::set<unsigned> >* који представља скупове кандидата за сваку колону матрице A. Метода враћа објекат типа *std::vector<Matrix>* који представља скуп кандидата за матрицу A. Идеја методе је да се прво генерише матрица преко скупова кандидата колона, а затим се провери да ли је матрица регуларна израчунавањем детерминанте. Уколико је добијена матрица регуларна, проверава се да ли све врсте генерисане матрице постоје у скуповима кандидата врста. Ако је и овај услов испуњен, матрица се додаје у скуп кандидата. Уколико бар један од ова два услова није испуњен, текућа матрица се одбацује и генерише се нова.

Уместо провере детерминанте могуће је проверавати да ли је нека колона линеарна комбинација било којих других колона, али се тај приступ показао нефикасним због сталне потребе за чувањем линеарних комбинација.

Генерисање кандидата за вектор b

Прототип метода за генерисање кандидата за вектор b

```
std::vector< std::set<unsigned> > BooleanFunction::
getCandidatesForVectorB(BooleanFunction f,
BooleanFunction g,std::vector<Matrix> candidatesA);
```

Метода као аргументе прихвата два објекта два објекта типа *BooleanFunction* и један објекат типа *td::vector<Matrix>* који редом представљају функције f и g и скуп кандидата за матрицу A. Метод враћа објекат типа *std::vector< std::set<unsigned> >* који представља скупове кандидата вектора b за сваку матрицу кандидата. Као инваријанта користи се расподела скупа апсолутних вредности Волшовог спектра. Током извршавања позива се метода за израчунавање суседних функција за функције f и g по задатим векторима. Затим се израчунавју Волшови спектри добијених суседних функција и пореде се расподеле скупова апсолутних вредности Волшових спектара. Уколико се расподеле поклопе, за сваку матрицу кандидата израчунава се кандидат за вектор b.

Генерисање кандидата за вектор l

Прототип метода за генерисање кандидата за вектор l

```
std::vector< std::set<unsigned> > BooleanFunction::
getCandidatesForVectorL(std::vector<int> fWalsh,
std::vector<int> gWalsh, unsigned vNum, unsigned
valNum,std::vector<Matrix> candidatesA);
```

Метода као аргументе прихвата два објекта два објекта типа *std::vector<int>* који редом представљају Волшове спектре функција f и g, два неозначена цела броја који представљају дужину вектора и арност функције, и овјекат типа *td::vector<Matrix>* који представља кандидате матрица. Метод враћа објекат типа std::vector< std::set<unsigned> > који представља скупове кандидата вектора l за сваку матрицу кандидата. Као инваријанта користи се расподела скупа апсолутних вредности Волшовог спектра. Метода пореди апсолутне вредности Волшових спектара функција f и g. За свако поклапање вредности Волшових спектара израчунава се кандидат за вектор l за сваку матрицу кандидата.

4.1.3 Метод за утврђивање афине еквивалентности функција

У овој тачки дат је опис метода која одје одговор да ли су задате Булове функције афино еквивалентне или не. Прототип метода за утврђивање да ли су функције афино еквивалентне:

```
void BooleanFunction::checkAffinelyEquivalence
(BooleanFunction f, BooleanFunction g,
std::ostream& ostr)
```

Метода као аргументе прихвата два објекта два објекта типа *BooleanFunction* који представљају функције f и g и објекат типа *std::ostream& ostr* који показује на место где ће бити исписан резултат. Ова метода састоји се од позива претходно описаних метода. Прво се израчунавају Волшови и аутокорелациони спектри задатих Булових функција. Као инваријанта користе се пасподеле апсолутних вредности Волшовог и аутокорелацоног спектра. Пореде се одговарајуће расподеле скупова апсолутних вредности Волшових и аутокорелационих спектара. Ако расподеле нису једнаке, издаје се порука да функције нису афино еквивалентне. Ако су расподеле једнаке, генеришу се кандидати за врсте и колоне матрице A. Потом се проверава се да ли скупови кандидата врста имају пресеке и ако немају, шаље се порука да функције нису афино еквивалентне. Потом се позивају методи за генерисање кандидата за матрицу A и векторе b и l. На крају се за сваку тројку матрице и два њој одговарајућа вектора провери да ли су функције афино еквивалентне и ако јесу, тројка A, b, l се прослеђује на излаз.

4.2 Програмска реализација и резултати тестова

У овој тачки изложени су тестови и методи за генерисање тестова алгоритма за проверу афине еквивалентности двеју задатих Булових функција. У другом делу дат је преглед резултата тестирања.

4.2.1 Тестови и методи за генерисање тестова

За тестирање алгоритма коришћено је по 10 насумично одабраних парова функција из $R(s, n)$ за различите s и n. Прва функција бира се насумично, док се друга функција генерише од прве помоћу случајно одабране регуларне матрице и два случајно одабрана вектора. За насумично генерисање функција из $R(s, n)$ користи се матрица генератор за Рид Малерове кодове реда n. У наставку су описани методи за насумично генерисање

матрица и функција. Детаљан преглед метода дат је у датотекама *BooleanFunction.cpp* и *matrix.cpp*.

Метод за генерисање матрице генератора за Рид Малерове кодове реда n

Прототип метода за генерисање матрице генератора за Рид Малерове кодове реда n:

```
Matrix Matrix::generateReedMullerMatrix
(unsigned n, std::vector<std::string>& monomials)
```

Метода на улазу прихвата неозначен цео број који представља арност функције и одређује дужину кода. Други аргумент је референца на вектор у који ће се смештати знаковне ознаке производа. Метода враћа објекат типа *Matrix* који представља матрицу генератора за Рид Малерове кодове реда n. Матрица се генерише попуњавањем врста. Прво се прва врста постави на јединичну функцију. Потом се наредних n врста попуњава векторима $x_1, x_2, \ldots x_n$. Затим се попуњава наредних $\binom{n}{2}$ врста свим производима дужине 2, па наредних $\binom{n}{3}$ врста свим производима дужине 3 итд. Ово попуњавање је реализовано тако да се $\binom{n}{k}$ врста попуњава као производ првих n и последњих $\binom{n}{k-1}$ врста. Приликом израчунавања производа чува се његова ознака.

Метода за насумично генерисање Булових функција из $R(k, n)$

Прототип метода за насумично генерисање Булових функција из $R(k, n)$:

```
BooleanFunction BooleanFunction::randomReedMuller
KNBooleanFunction(unsigned k, unsigned n,
std::vector<std::string>& anfFunction)}
```

Метода на улазу прихвата два неозначена цела броја који представљају арност n и ред функције k. Други аргумент је референца на знаковни низ у коме ће бити сачувана знаковна репрезентација функције. Ова метода прво позива метод за генерисање матрице генератора Рид Малерових кодова реда n. Потом се за сваки степен члана $s, s \leq k$ на случајан начин бира број чланова степена s. Када се одабере број чланова, на случајан начин бирају се и сами чланови полинома. По избору сваког члана чува се и његова знаковна репрезентација. Случајно генерисана функција представљена је збиром случајно одабраних чланова полинома.

Табела 4.1: Резултати тестирања алгоритма $R(2,4)$ до $R(4,4)$

	$R(2,4)$		$R(3,4)$		$R(4,4)$	
	ФА	време у мс	ФА	време у мс	ФА	време у мс
1	34500	860	29580	220	50ee0	200
2	4ea3e0	9634	2e80	30	49d250	5761
3	34e00	410	2e80	30	59e80	540
4	4eaf20	6912	2fa0	20	4ca2e0	522
5	4ea3e0	4651	29ac0	180	4e0a20	6508
6	4eaf20	3341	2f40	20	4ca250	3530
7	22e00	170	2f40	20	49d220	3791
8	4ce00	260	2f40	30	35ee0	160
9	34e00	200	29a00	170	4e9b40	3740
10	bfe00	631	2f40	20	4ca250	3531

4.2.2 Резултати тестирања

Овде су изложени резултати тестирања алгоритма за проверу афине еквивалентности двеју задатих Булових функција. Тестирање је спроведено над класама Булових функција арности од 4 до 11 узимајући редом све класе облика $R(k,n), k = \overline{2,11}, n = \overline{4,11}$. За сваку насумично генерисану функцију бележено је време извршавања алгоритма и број провера тројки кандидата A, b и l које нису афина тфансформација прве у другу функцију (број лажних аларма). Резултати добијени програмом представљени су у табели 4.1. За комплетне резултате и детаље о генерисаним функцијама погледати датотеке дате у прилогу.

Приликом тестирања функција из $R(k,n)$ за различите вредности k и n уочено је да алгоритам брже долази до решења када је разлика између k и n мала. Из резултата се види да је тада број кандидата матрица знатно мањи него када је разлика између k и n велика. Тестирање је показало да је број кандидата матрица највећи за функције малог степена и велике арности. Ово је и очекивано, јер ја тада вероватноћа генерисања бент функције већа него у осталим случајевима, а што је функција сличнија бент функцији то је број кандидата већи. Алгоритам почиње да успорава за арности функције $n > 10$ и тада није практично користан. За арности функције $n \leq 10$ може се практично искористити као алат за решавање проблема класификације или набрајања Булових функција неких интересантних скупова Булових функција.

Глава 5

Закључак

У раду је реализован алгоритам за проверу афине еквивалентности две задате Булове функције занован на примени неколико инваријанти. Практично је применљив за проверу афине еквивалентности Булових финкција које зависе од око 10 променљивих. Алгоритам се у овом облику може користити као алат за класификацију неких скупова Булових функција малог степена. За унапређење алгоритма неопходно је детаљно изучавање класа Булових функција и проналажење што већег броја трансформација Булових функција и инваријанти.

Циљ овог рада није само представљање једног алгоритма за проверу афине еквивалентности функција. Рад треба да помогне у разумевању значаја проблема што би могло да доведе до проналажења ефикаснијих решења.

Литература

[1] Q. Meng, M. Yang, H. Zhang and Y. Liu. *Analysis of affinely equivalent Boolean functions.* Science in China Series F: Information Sciences, Volume 50, Issue 3, pp 299-306, Science in China Press 2007.

[2] T. Cusick, P. Stanica. *Cryptographic Boolean Functions and Applications*, Academic Press, 2009.

[3] MacWilliams, Sloane. *The theory of error-correcting codes*, North-Holland Publishing Company, 1978.

[4] E. Brier, P. Langevin *Classification of Boolean cubic forms in nine variables.* Proceedings of IEEE Information Theory Workshop, 2003. 179-182

[5] Preneel B. *Analysis and design of cryptographic hash functions.* Ph. D Thesis, Leuven (Belgium): Katholieke University, 1993.

[6] J. Fuller, W. Millan. *Linear redundancy in S-box.* Fast Software Encryption, Springer-Verlag, 2003. 74-86