UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika - uporabna smer (UNI)

Tadej Novak

PREVERJANJE NAKLJUČNOSTI V KRIPTOGRAFIJI

Diplomsko delo

Ljubljana, 2000

Kazalo

PROGRAM DIPLOMSKEGA DELA

Preverjanje naključnosti v kriptografiji

Delo naj predstavi matematične osnove, potrebne za študij statističnih lastnosti kriptografskih algoritmov kot so simetrični in tekoči algoritmi. Glavni cilj je predstaviti osnovne principe, ki se uporabljajo pri današnjih statističnih testiranjih za oceno naključnosti in s tem varnosti šifrirnih algoritmov in generatorjev psevdonaključnih števil. Izdela naj se tudi matematična utemeljitev nekaterih osnovnih statističnih testov generatorjev naključnih števil.

Mentor: A. Jurišić

Somentor: M. Perman

Literatura:

SUSAN LANDAU, *Communications Security for the Twenty-first Century: The Advanced Encryption Standard*, Notices of the AMS, **47** (2000), 450–459.

D. R. STINSON, *Cryptography, Theory and Practice*, CRC Press, 1995.

A. MENEZES, P. VAN OORSCHOT AND S. VANSTONE, *Handbook of Applied Cryptography*, Series on Discrete Mathematics and its Applications, 4th ed., CRC Press, 1999.

R. JAMNIK, *Matematična statistika*, DZS, Ljubljana, 1980.

Povzetek. Delo vključuje tri področja matematične znanosti, ki so potrebna za razumevanje preverjanja naključnosti generatorjev naključnih zaporedij bitov. To so: kriptografija, verjetnostni račun in statistika. Delo najprej opiše, kako lahko zgeneriramo psevdonaključno zaporedje bitov z lepimi statističnimi lastnostmi (LFSR generator) in na kakšen način ga potem lahko uporabimo v kriptografiji. Nato predstavi osnovno znanje verjetnostnega računa in statistike, ki je potrebno za razumevanje ne le rezultatov, ampak tudi lastnosti posameznih statističnih testov, ki so opisani v četrtem poglavju. To so testi psevdonaključnih generatorjev bitov, ki poleg osnovnih statističnih lastnosti, ki naj bi jih imelo izhodno zaporedje generatorja, preverjajo tudi, ali je iz tega psevdonaključnega zaporedja možna rekonstrukcija semena.

Abstract. This dissertation includes three fields of mathematical science, which are needed to understand random bit generators' randomness testing. These are: cryptography, probability theory and mathematical statistics. In the beginning, this paper describes how to generate pseudorandom bit sequence with good statistical properties (LFSR generator) and how to use it in cryptography. Next, it presents the basic knowledge of probability theory and statistics, which is needed to understand not only the results, but also properties of statistical tests, which are described in the fourth chapter. These pseudorandom bit generators' tests verify basic statistical properties, which an output sequence generator should have, as well as establish the possibility of seed reconstruction from this pseuderandom sequence.

Math. Subj. Class. (2000): 60F05, 6204, 62G10, 68P25, 94A24, 94A55, 94A60

Ključne besede: kriptografija, varnost, razbijanje kriptosistemov, napadi, tokovni tajnopis, psevdonaključno, naključno zaporedje bitov, LFSR, Diehard, Crypt-X, verjetnostni račun, matematična statistika, statistični test, neparametrični testi, hi kvadrat test, test Kolmogorova.

Key words: cryptography, security, cryptoanalysis, attacks, stream cyphers, pseudorandom, random bit sequence, LFSR, Diehard, Crypt-X, probabylity theory, mathematical statistics, statistical test, nonparametric tests, chi square test, Kolmogorovs' test.

UVOD

Ljudje si že od nekdaj izmenjujejo informacije. Najvarnejša izmenjava je pogovor na "štiri oči", ker pa to ni vedno mogoče, smo si omislili kurirje, telegraf, telefon, pošto, radijske postaje, elektronsko pošto, internet ipd. Prenosnik informacij mnogokrat ni najbolj zanesljiv, vedno ga lahko kdo opazuje in na nek način od njega izve, kakšne informacije prenaša. To pa je razlog, zakaj se je začel razvoj šifrirnih sistemov, ki je doživel največji razmah v 20. stoletju, predvsem s civilno uporabo interneta. Dejavnikov, ki vplivajo na razvoj kriptografije, je več, v grobem pa bi jih lahko razdelili na dve skupini. V prvi so dejavniki, ki predstavljajo fizično zgradbo sistema:

1. **matematika** (teorija števil, ...),
2. **računalništvo** (analiza algoritmov) in
3. **elektrotehnika** (hardware).

V drugi skupini pa so tisti, ki narekujejo sam razvoj:

1. **uporabniki aplikacij** (finance, ...),
2. **politika** (prepoved izvoza kriptosistemov, ...),
3. **pravo** (patenti, podpisi, jamstva, ...),
4. **družba** (želja po zasebnosti),

in bi jih lahko označili kot uporabnike kriptosistemov.

Za oceno varnosti kriptosistema je potrebno preveriti predvsem, ali obstaja način, da kot opazovalec pretoka informacij na nek način pridemo do le teh. Načinov, kako to preverimo, pa je več. Najpreprostejši je požrešni napad, s katerim preskusimo vse ključe. Da je algoritem, ki se uporablja dandanes, odporen proti temu napadu, naj bi uporabljal vsaj 80-bitni ali celo 90-bitni ključ, saj je preskus v povprečju polovice od $2^{80} \approx 10^{24}$ ključev prezamuden za stanje tehnike, kakršno bo predvidoma v naslednjih 20 letih. Tu upoštevamo **Moorovo načelo**, ki pravi, da se hitrost, ki jo zmore dana tehnologija, podvoji vsakih 18 mesecev. To pomeni, da se dolžina ključa za zagotovitev iste stopnje varnosti vsakih 18 mesecev poveča za en bit. Drugi način je, da poiščemo slabost algoritmov. Lahko so to linearne operacije, ki jih uporablja, povezava med posameznimi deli tajnopisa ipd. Pri razbitju DES-a[1] je bil najuporabnejši pristop t.i. diferenčna kriptoanaliza, za opis glej [18, str. 98–104].

V letošnjem letu se bo vlada ZDA standardizirala nov algoritem za šifriranje. Imenoval se bo Advanced Encription Standard (AES) in bo pravzaprav naslednik DES-a. Izbran bo kot najboljši izmed 21 predlaganih na podlagi zahtevanih kriterijev, kot so hitrost, varnost, prilagodljivost, prostorska zahtevnost ipd. Bistvena novost pa je ta, da so bili predlogi sprejeti s celega sveta in da pri ocenjevanju sodelujejo tudi mednarodni strokovnjaki, ki niso iz ZDA. Ameriški agenciji za varnost (NSA - National Security Agency) in s tem ameriški politiki, ki uvršča kriptografijo med vojaško orožje, to najbrž ni najbolj všeč, a morali so se sprijazniti z dejstvom, da so najuspešnejše napade na DES naredili tuji strokovnjaki, predvsem izraelski in japonski.

[1] Data Encription Standard

Dobri šifrirni sistemi morajo biti hkrati tudi dobri generatorji psevdonaključnih števil. Njihovo naključnost pa lahko preverimo s statističnimi testi (osnovni statistični testi). Morajo pa biti odporni tudi na že znane napade, kar preverjajo bolj zahtevni statistični testi, ki jih bomo opisali v 4. poglavju. Velik poudarek bo na preskusu nelinearnosti.

Diplomsko delo je razdeljeno na štiri poglavja:

1. **poglavje** nam na kratko poda nekaj splošnih dejstev in definicij iz kriptografije. Predstavi nam nekaj najbolj znanih šolskih primerov šifrirnih algoritmov in njihova razbitja na podlagi statističnih lastnosti tajnopisa. Opredeli tudi, kakšni so kriptografsko varni generatorji psevdonaključnih števil in v zadnjem razdelku poda primer generatorja LFSR, ki ima zelo dobre osnovne statistične lastnosti in se ga da dobro implementirati v strojni opremi. Dani so tudi osnovni napotki, na kaj je potrebno paziti, da bo kriptografsko varen.

2. **poglavje** nam da osnove verjetnostnega računa. Ukvarja se s slučajnimi spremenljivkami in njihovimi porazdelitvami, največji poudarek pa je na limitnih porazdelitvah zaporedja slučajnih spremenljivk. Te so podlaga mnogih statističnih testov.

3. **poglavje** opisuje področje matematične statistike. Interpretacija rezultatov statističnih testov je mnogokrat napačna, k čemur prispeva nerazumevanje pojmov, kot so moč testa, stopnja značilnosti testa ipd. Ti pojmi so podrobno opisani, na koncu poglavja pa sta dana tudi dva neparametrična testa, ki sta najpogostejša pri statističnih testiranjih generatorjev psevdonaključnih števil.

4. **poglavje** vsebuje testne funkcije programov za testiranje naključnosti generatorjev naključnih zaporedij bitov, ki se danes največ uporabljajo in ki jih potrebujemo tudi za testiranje šifrirnih algoritmov. Opisana sta programa Crypt-XS in Diehard.

Na začetku vsakega poglavja je uvod, ki opisuje namen poglavja, za tistega, ki bi rad vedel več, pa tudi, kje najde nadaljnjo literaturo.

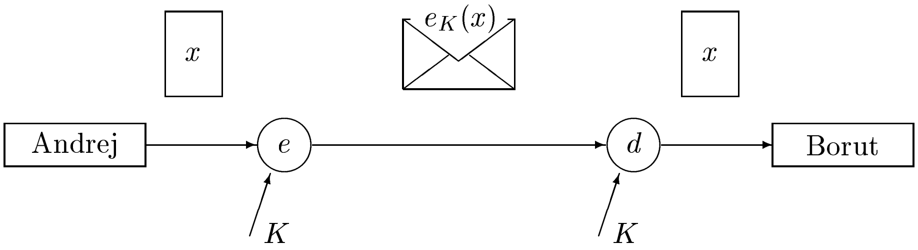
Za neljube napake, ki so ostale v delu, se bralcu opravičujem. Vse, ki jih bom uspel najti, pa bom objavil na spletni strani
`http://valjhun.fmf.uni-lj.si/~ajurisic/diplome/novak_p.html`.

1 VARNOST KRIPTOSISTEMOV

Prvi trije razdelki podajajo osnovno znanje o kriptografiji in so povzeti po Stinsonovem učbeniku [18]. Prvi razdelek nam opisuje, kaj sploh je kriptosistem in kaj pomeni, da smo ga razbili. Največji občutek zadovoljstva pa dobimo, ko sami razbijemo kak kriptosistem. Kako storimo to na dveh preprostih tajnopisih, substitucijskem in Vigenèrejevem, je opisano v drugem razdelku. Kaj si lahko predstavljamo pod pojmom kriptografska varnost, je intuitivno opisano v prvem razdelku, v tretjem je podanih tudi več definicij, ki opisujejo različne poglede na varnost in stopnjo varnosti. Zadnja dva pa nam opisujeta, kako lahko naredimo in uporabljamo psevdonaključne generatorje bitov v kriptografiji. Opisan je LFSR generator ter kaj moramo storiti, da postane kriptografsko varen. V [15, str. 216–222] najdemo tudi nadaljnje reference za bolj podrobno razumevanje matematičnega ozadja LFSR in njegovih možnosti za uporabo.

1.1 STOPNJE VARNOSTI KRIPTOSISTEMOV

Osnovni cilj kriptografije je omogočiti dvema osebama, recimo Andreju in Borutu, komuniciranje preko neke povezave, npr. e-pošte, na tak način, da opazovalec, npr. Oskar, ne bi mogel razumeti, kakšne informacije si izmenjujeta. Tekst x, ki ga Andrej želi poslati Borutu, bomo imenovali čistopis. To je lahko običajen tekst, lahko pa so računalniške datoteke slik, zvoka ipd. S prej dogovorjenim ključem K, ki ga poznata le Andrej in Borut, bo Andrej sporočilo zašifriral s postopkom e_K in ga nato poslal Borutu. Zašifrirano sporočilo imenujemo tajnopis in ga Borut lahko dešifrira s pomočjo dešifrirnega postopka d_K prirejenega ključu K.



Slika 1: Shema kriptosistema

Ker Oskar ne pozna ključa K, pravimo, da je naš postopek **kriptosistem s privatnim ključem** in ga opišemo kot peterico $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ za katero velja:

1. $\mathcal{P}$ je končna množica možnih **čistopisov** (plaintext),
2. $\mathcal{C}$ je končna množica možnih **tajnopisov** (ciphertext),
3. $\mathcal{K}$ je končna množica možnih **ključev** in
4. za vsak ključ $K \in \mathcal{K}$ imamo natanko določen **šifrirni postopek** $e_K \in \mathcal{E}$ (encryption rule) in ustrezen **dešifrirni postopek** $d_K \in \mathcal{D}$ (decryption rule), kjer sta $e_K : \mathcal{P} \to \mathcal{C}$ in $d_K : \mathcal{C} \to \mathcal{P}$ taki funkciji, da je $d_K(e_K(x)) = x$ za vsak čistopis $x \in \mathcal{P}$.

Najpomembnejša je četrta točka, ki pravi, da ko Borut dešifrira sporočilo $e_K(x)$, mora dobiti prvotno sporočilo x, ki mu ga je poslal Andrej. Slika 1 nam prikazuje celotno shemo delovanja kriptosistema.

V splošnem velja kriptosistem za varnega, če Oskar, kljub temu da pozna delovanje sistema, ne more ugotoviti ključa, s katerim je bilo sporočilo šifrirano. V tem primeru pravimo, da je kriptosistem **varen po Kerckhoffovem principu**. Za kriptosisteme, ki so danes v komercialni uporabi, to pomeni, da ne obstaja metoda, ki bi za cel velikostni razred hitreje, kot je preskus vseh možnih ključev, ugotovila dejanski ključ sporočila. Če z n označimo dolžino zapisa ključa, je velikost prostora ključev navadno enaka 2^n, zato preiskus vseh ključev v povprečju traja $2^n/2$ časovnih enot. Če sedaj s $T(n)$ označimo še število časovnih enot, potrebnih za ugotovitev dejanskega ključa pri neki metodi, potem to, da je kriptosistem varen, pomeni, da ne obstaja metoda, pri kateri bi bila limita $\lim_{n\to\infty} T(n)/2^n$ enaka $\mathcal{O}\left((1/2^n)^r\right)$ za neko pozitivno število r. Glede na podatke, ki jih dobi napadalec Oskar za ugotavljanje ključa, lahko napade na kriptosistem razdelimo na več težavnostnih stopenj:

1. **samo tajnopis**: Oskar pozna tajnopis x,
2. **poznani čistopis**: Oskar pozna čistopis x in ustrezen tajnopis $y = e_K(x)$,
3. **izbrani čistopis**: Oskar ima začasno na voljo šifrirni stroj (ki uporablja funkcijo e_K, ki je Oskar ne pozna) in za poljubno izbran čistopis x lahko določi ustrezen tajnopis $y = e_K(x)$,
4. **izbrani tajnopis**: Oskar ima začasno na voljo dešifrirni stroj (ki uporablja funkcijo d_K, ki je Oskar ne pozna) in za poljubno izbran tajnopis y lahko določi ustrezen čistopis $x = d_K(y)$.

Za varen prenos velikih količin podatkov potrebujemo kriptosistem, ki je odporen proti vsem zgoraj naštetim stopnjam napada.

1.2 RAZBIJANJE PREPROSTIH KRIPTOSISTEMOV

Snov je povzeta po Stinsonovem učbeniku [18, sect. 1.2], kjer so podani tudi konkretni primeri šifer ter postopki, ki jih je avtor uporabil za njihovo razbitje.

1.2.1 Substitucijski tajnopis

Denimo, da hočemo šifrirati sporočilo napisano samo z malimi črkami (presledke spustimo) in v angleškem jeziku. Šifriramo tako, da vsaki črki priredimo novo črko, npr.

$$a \to B, \quad b \to C, \quad \ldots \quad , y \to Z, \quad z \to A\,,$$

ločila in presledke pa izpustimo, saj slednji nastopajo in v slovenskem in v angleškem jeziku z relativno frekvenco skoraj 20%, njihova uporaba pa besedilo hitro razdeli na posamezne besede. Če črke spremenimo v ustrezne številke ($a \to 0$, $b \to 1$, …), lahko podamo še matematičen opis tega kriptosistema.

Naj bo $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$ in naj $\mathcal{K}$ vsebuje vse možne permutacije števil $0, 1, \ldots, 25$. Za vsako permutacijo $\pi \in \mathcal{K}$ definirajmo

$$e_\pi(x) = \pi(x) \qquad \text{in} \qquad d_\pi(y) = \pi^{-1}(y)\,,$$

kjer je π^{-1} inverz permutacije π.

Velikost prostora ključev je $26! \approx 2^{88}$. Če bi bil edini način za razbitje preverjanje vseh ključev, potem bi bil ta kriptosistem dovolj varen glede na današnje razmere. Glej poročilo skupine kriptografov in računalničarjev, ki se nahaja na
`http://valjhun.fmf.uni-lj.si/~ajurisic/seminar/DES/keylengths.txt`
Toda ta tajnopis ne zagotavlja nikakršne varnosti, kar bomo tudi pokazali. Tajnopis je seveda

brezpogojno varen, če šifriramo le eno črko, vendar pa ponavadi z istim ključem šifriramo besedilo dolgo vsaj 5 vrstic, kar je povsem dovolj za razbitje.

Postopek za razbitje je preprost. V tajnopisu le pogledamo, katere črke se ponovijo največkrat, in potem s pomočjo Tabele 1 ugibamo, na kakšen način so bile črke zamenjane. Tabela je povzeta iz [2, str. 395–400] za angleški jezik in iz [7] za slovenski jezik in je bila narejena z vzorcem teksta iz različnih knjig, revij in časopisov. V omenjeni literaturi so tudi tabele relativnih frekvenc, kateri pari in katere trojke črk nastopajo v tekstih skupaj. To pa je dovolj, da naredimo program, ki nam s pomočjo vgrajenega majhnega slovarčka besed iz tajnopisa poišče čistopis. Torej obstaja razbitje kriptosistema na najtežjem nivoju – "samo tajnopis".

Slovenski jezik

črka	verjetnost	črka	verjetnost
E	0'115	D	0'028
A	0'103	P	0'025
I	0'095	Z	0'025
O	0'088	U	0'024
N	0'084	B	0'019
R	0'079	G	0'013
S	0'077	Č	0'010
L	0'076	H	0'003
J	0'050	Š	0'001
T	0'035	C	0'001
V	0'035	Ž	0'001
K	0'030	F	0'000
M	0'029		

Angleški jezik

črka	verjetnost	črka	verjetnost
E	0'127	M	0'024
T	0'091	W	0'023
A	0'082	F	0'022
O	0'075	G	0'020
I	0'070	Y	0'020
N	0'067	P	0'019
S	0'063	B	0'015
H	0'061	V	0'010
R	0'060	K	0'008
D	0'043	J	0'002
L	0'040	Q	0'001
C	0'028	X	0'001
U	0'028	Z	0'001

Tabela 1: Relativne frekvence črk v slovenskem in angleškem jeziku

1.2.2 Vigenèrejev tajnopis

Kriptosistem izvira iz 16. stoletja, zapisal pa ga je Blaise de Vigenère. Tako kot pri substitucijskem tajnopisu bomo tudi tu črke angleške abecede spremenili v številke, ključ pa nam bo predstavljala neka m črk dolga beseda, ki jo bomo prištevali besedilu. Pokažimo delovanje kar na primeru, ko bi s ključem key radi zašifrirali besedilo $information$:

$$\begin{array}{r} information \\ +\ keykeykeyke \end{array} \sim \begin{array}{rrrrrrrrrrrr} & 8 & 13 & 5 & 14 & 17 & 12 & 0 & 19 & 8 & 14 & 13 \\ + & 10 & 4 & 24 & 10 & 4 & 24 & 10 & 4 & 24 & 10 & 4 \\ & - & - & - & - & - & - & - & - & - & - & - \\ & 18 & 17 & 3 & 24 & 21 & 10 & 10 & 23 & 6 & 24 & 17 \end{array} \sim RQCXUJJWFXQ\,.$$

Podajmo še matematičen opis tega kriptosistema.

Naj bo $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$. Za dan ključ $K = (k_1, \ldots, k_m)$ definirajmo

$$e_K(x_1, \ldots, x_m) = (x_1 + k_1, \ldots, x_m + k_m) \quad \text{in} \quad d_K(y_1, \ldots, y_m) = (y_1 - k_1, \ldots, y_m - k_m)\,,$$

kjer vse operacije potekajo v $\mathbb{Z}_{26}$, torej po modulu 26.

Velikost prostora ključev je 26^m, recimo pri $m = 15$ to pomeni približno $\approx 2^{70}$, kar naj bi pri današnjih zahtevah dalo dokaj visok nivo varnosti. Toda tudi ta tajnopis, četudi Oskar ne pozna niti dolžine m ključa, ne zagotavlja nobene varnosti, če z istim ključem zašifriramo vsaj 10 vrstic nekega besedila.

Razbitje bomo naredili s pomočjo indeksa naključja, kot je to naredil Wolfe Friedman leta 1920. **Indeks naključja** $I_C(\boldsymbol{x})$ je definiran kot verjetnost, da sta v danem nizu črk $\boldsymbol{x}$ dve poljubno izbrani črki enaki. Če vrednosti v tabeli 1 označimo s p_i, kjer je p_i relativna frekvenca za i-to črko abecede, potem je indeks naključja za angleški tekst enak

$$I_C \approx \sum_{i=0}^{25} p_i^2 = 0,065\,,$$

za razliko od povsem naključno izbranega zaporedja črk[2], kjer je

$$I_C = \sum_{i=0}^{25} (1/26)^2 = 0,038\,.$$

Če za dan niz $\boldsymbol{x} = x_1 \ldots x_n$ zaporedoma označimo z $f_0, f_1, \ldots f_{25}$ število pojavitev črk A, B, ... , Z, potem je indeks naključja za ta niz enak

$$I_C(\boldsymbol{x}) = \sum_{i=0}^{25} \binom{f_i}{2} \Big/ \binom{n}{2}\,.$$

Pokazali bomo, kako Oskar iz danega tajnopisa ugotovi dolžino ključa. Izkoristili bomo dejstvo, da če vsaki črki angleškega teksta prištejemo črko z vrednostjo $r \in \mathbb{Z}_{26}$, se indeks naključja ne spremeni. Recimo, da je bilo besedilo $x_1 x_2 x_3 \ldots$ zakodirano s ključem dolžine 4. Potem morajo biti vsi indeksi $I_C(x_1 x_5 x_9 \ldots)$, $I_C(x_2 x_6 x_{10} \ldots)$, $I_C(x_3 x_7 x_{11} \ldots)$ in $I_C(x_4 x_8 x_{13} \ldots)$ enaki približno 0'065. Če pa je dolžina ključa različna od 4, potem bodo ti indeksi enaki približno 0'038. Kako nadaljujemo z dešifriranjem tajnopisa, je opisano v Stinsonovem učbeniku [18, str. 33-36].

1.3 BREZPOGOJNA IN RAČUNSKA VARNOST

V tem razdelku bomo predstavili dva osnovna pogleda na varnost kriptosistemov, kot jo je predstavil Claude Shannon leta 1949, mi pa jo bomo povzeli iz [18, str. 44–51]. Predstavil je tudi uspešen napad tipa "samo tajnopis", če za šifriranje vseskozi uporabljamo isti ključ. Napad je opisan v [18, str. 51–63], vendar pa se bomo mi raje osredotočili na proučevanje varnosti pri pogoju, da dan ključ uporabimo le enkrat.

Za dan kriptosistem pravimo, da je pri dani stopnji napada **računsko varen**, če najboljši algoritem za razbitje opravi vsaj N operacij, kjer je N neko predpisano veliko število in se nanaša na zmogljivost današnjih računalnikov. V praksi nismo tako formalni in pravimo, da je kriptosistem **računsko varen**, če najboljši znan algoritem za razbitje opravi vsaj N operacij, vendar pa to še zdaleč ni dolgoročno zagotovilo za varnost. Veliko današnjih kriptosistemov pa je takih, da obstaja njihovo razbitje le, če obstaja algoritem za rešitev nekega težkega matematičnega problema. To so navadno t.i. NP-polni problemi, znani 100 in več let, glej

M. PETKOVŠEK, T. PISANSKI, *Izbrana poglavja iz računalništva, 1. del: Izračunljivost in rešljivost, jeziki, NP-polnost, naloge*, DMFA SR Slovenije, Ljubljana, 1986.

Na drugi strani pa so **brezpogojno varni** kriptosistemi, ki jih pri dani stopnji napada ni mogoče razbiti niti z neomejeno zmogljivimi računalniki. Glede na to, kakšen nivo varnosti zagotavlja avtor kriptosistema, lahko definiramo pojem **razbitja**. V drugem primeru je to lahko tudi preskus vseh ključev, četudi ga je z današnjo tehniko nemogoče opraviti v kratkem času.

Opišimo sedaj, kaj pomeni, da je dan kriptosistem brezpogojno varen pri napadu s poznanim tajnopisom. V tem primeru je razbitje poljuben postopek, ki nam za dan tajnopis y da nek

[2] V angleščini je za tak tekst pogosta fraza "monkey written text".

čistopis x, za katerega zagotavlja, da je verjetnost, da je x rezultat dešifriranja tajnopisa y pri iskanem ključu K, večja od verjetnosti, da za poljubno šifriranje izberemo čistopis x, skratka

$$P_{\mathcal{P}}(x|y) > P_{\mathcal{P}}(x)\,.$$

Zato pravimo, da kriptosistem zagotovlja **popolno tajnost**, če je

$$P_{\mathcal{P}}(x|y) = P_{\mathcal{P}}(x)$$

za vsak par $x \in \mathcal{P}$ in $y \in \mathcal{C}$, kjer je $y = e_K(x)$. Torej nam pri nobenem poskusu razbitja ne pomagajo niti frekvence posameznih črk niti kakršnokoli drugo orodje. S stališča verjetnostnega računa zadnja enačba pomeni, da sta slučajni spremenljivki x in y neodvisni.

Trditev 1.3.1. *Za poljuben čistopis $x \in \mathcal{P}$ označimo s $P_{\mathcal{P}}(x)$ verjetnost, da je izbran za šifriranje,[3] in za poljuben ključ $K \in \mathcal{K}$ je $P_{\mathcal{K}}(K)$ verjetnost, da ga uporabimo kot ključ šifrirne funkcije. Potem je verjetnost, da je dan tajnopis y dobljen s šifriranjem čistopisa x, enaka*

$$P_{\mathcal{P}}(x|y) = \frac{P_{\mathcal{P}}(x) \displaystyle\sum_{\{K\,;\,x=d_K(y)\}} P_{\mathcal{K}}(K)}{\displaystyle\sum_{\{K;y\in C(K)\}} P_{\mathcal{K}}(K)P_{\mathcal{P}}(d_K(y))}\,,$$

kjer smo s $C(K)$ označili množico

$$C(K) = \{e_K(x)\,;\, x \in \mathcal{P}\}\,.$$

Dokaz. Verjetnost, da smo pri izbranem šifriranju dobili tajnopis y, je enaka

$$P_{\mathcal{C}}(y) = \sum_{\{K\,;\,y\in C(K)\}} P_{\mathcal{K}}(K)P_{\mathcal{P}}(d_K(y))\,. \tag{1}$$

Verjetnost, da iz danega čistopisa x dobimo tajnopis y, pa je odvisna le od izbora ključa

$$P_{\mathcal{C}}(y|x) = \sum_{\{K\,;\,x=d_K(y)\}} P_{\mathcal{K}}(K)\,.$$

Če dobljeni enačbi vstavimo v Bayesov obrazec

$$P_{\mathcal{P}}(x|y) = \frac{P_{\mathcal{P}}(x)P_{\mathcal{C}}(y|x)}{P_{\mathcal{C}}(y)}\,,$$

dobimo željeno enačbo. ■

Naslednja trditev in izrek, ki ji sledi, bosta navedla preprosta pogoja, ki sta potrebna in tudi zadostna za zagotovitev popolne tajnosti v kriptosistemu pri najmanjšem možnem prostoru ključev. Iz dokaza izreka bomo lahko razbrali tudi, da je verjetnost $P_{\mathcal{C}}(y)$ enaka pri vsakem $y \in \mathcal{C}$, kar pomeni, da mora biti kriptosistem, ki zagotovlja popolno tajnost, dober generator naključnih izborov elementov množice $\mathcal{C}$. Primer takega kriptosistema je Vigenèrejev tajnopis.

Trditev 1.3.2. *Naj bo $P_{\mathcal{C}}(y) > 0$ za vsak $y \in \mathcal{C}$. Če kriptosistem vsebuje popolno tajnost, potem velja*

$$|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|\,.$$

[3] Če šifriramo tekst v slovenskem jeziku, je verjetnost, da šifriramo tekst `mati` gotovo večja od verjetnosti, da šifriramo tekst `abcd`.

Dokaz. Ker sistem vsebuje popolno tajnost, je $P_{\mathcal{P}}(x|y) = P_{\mathcal{P}}(x)$ za vsak $x \in \mathcal{P}$ in $y \in \mathcal{C}$, kar pa je po Bayesovem obrazcu ekvivalentno zahtevi $P_{\mathcal{C}}(y|x) = P_{\mathcal{C}}(y)$ za vsak $x \in \mathcal{P}$ in $y \in \mathcal{C}$. Pri fiksnem $x \in \mathcal{P}$ je torej pogoj $P_{\mathcal{C}}(y) > 0$ za vsak $y \in \mathcal{C}$ ekvivalenten pogoju $P_{\mathcal{C}}(y|x) > 0$ za vsak $y \in \mathcal{C}$. Ker nam poljuben ključ ne more dati iz danega čistopisa x dveh različnih tajnopisov in ker zadnji pogoj pravi, da lahko vsak tajnopis y dobimo iz izbranega čistopisa x, je ključev vsaj toliko kot možnih tajnopisov, torej $|\mathcal{K}| \geq |\mathcal{C}|$. Ker pa je vsaka šifrirna preslikava injektivna, je $|\mathcal{C}| \geq |\mathcal{P}|$. ∎

Pogoj $P_{\mathcal{C}}(y) > 0$ za vsak $y \in \mathcal{C}$ je ponavadi izpolnjen, saj če obstaja tajnopis $y \in \mathcal{C}$, za katerega je $P_{\mathcal{C}}(y) = 0$, ga lahko brez škode izločimo iz $\mathcal{C}$.

Izrek 1.3.3. *Naj bo $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ kriptosistem, za katerega velja $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}| = n$. Potem kriptosistem zagotavlja popolno tajnost, natanko tedaj ko je*
(1) vsak ključ izbran z enako verjetnostjo, to pomeni $P_{\mathcal{K}}(K) = 1/n$ za vsak $K \in \mathcal{K}$, in
(2) za vsak par $x \in \mathcal{P}$ in $y \in \mathcal{C}$ obstaja natanko en ključ $K \in \mathcal{K}$, za katerega velja $y = e_K(x)$.

Dokaz. Najprej privzemimo, da dan kriptosistem zagotavlja popolno tajnost. Izberimo si poljuben ključ K in ker mora biti e_K injektivna funkcija, je zaradi pogoja $|\mathcal{C}| = |\mathcal{P}|$ tudi bijektivna. Torej za vsak $y \in \mathcal{C}$ obstaja $x \in \mathcal{P}$, da je $y = e_K(x)$, kar pomeni, da je $P_{\mathcal{C}}(y) > 0$ za vsak $y \in \mathcal{C}$. Ker kriptosistem zagotavlja popolno tajnost, je potem tudi $P_{\mathcal{C}}(y|x) > 0$ za vsak $x \in \mathcal{P}$ in $y \in \mathcal{C}$, kakor smo to sklepali v dokazu prejšnje trditve. Če si sedaj izberemo še nek poljuben $x \in \mathcal{P}$, je potem $P_{\mathcal{C}}(y|x) > 0$ za ta $x \in \mathcal{P}$ in za vsak $y \in \mathcal{C}$. Torej za ta x in za vsak $y \in \mathcal{C}$ obstaja nek ključ $K \in \mathcal{K}$, da je $y = e_K(x)$. Iz tega sledi, da je za izbran x

$$\{e_K(x)\,;\, K \in \mathcal{K}\} \supset \mathcal{C}\,.$$

Če bi za nek $y \in \mathcal{C}$ bilo $y = e_K(x)$ pri dveh različnih ključih $K \in \mathcal{K}$, potem bi zaradi $|\mathcal{K}| = |\mathcal{C}|$ veljalo $|\{e_K(x)\,;\, K \in \mathcal{K}\}| < n$, kar pa, kot smo pokazali, ni mogoče.

Pokažimo sedaj $P_{\mathcal{K}}(K) = 1/n$. Indeksirajmo elemente v množici čistopisov $\mathcal{P} = \{x_1, \ldots x_n\}$. Izberimo si nek $y \in \mathcal{C}$ in s K_i, $i = 1, \ldots, n$, sedaj lahko označimo ključ, za katerega je $y = e_{K_i}(x_i)$, saj smo pokazali, da je tak ključ enolično določen. Tudi $P_{\mathcal{C}}(y|x_i) = P_{\mathcal{K}}(K_i|x_i)$ sledi iz tega, da je ključ enolično določen za vsak par y in x_i. In ker je izbor ključa K neodvisen od izbora čistopisa x_i, je $P_{\mathcal{C}}(y|x_i) = P_{\mathcal{K}}(K_i)$. Če slednje vstavimo v Bayesov obrazec

$$P_{\mathcal{P}}(x_i|y) = \frac{P_{\mathcal{P}}(x_i) P_{\mathcal{C}}(y|x_i)}{P_{\mathcal{C}}(y)}$$

in upoštevamo popolno tajnost $P_{\mathcal{P}}(x_i|y) = P_{\mathcal{P}}(x_i)$, dobimo, da za izbran y velja $P_{\mathcal{K}}(K_i) = P_{\mathcal{C}}(y)$ za vsak $i = 1, \ldots, n$. Torej je $P_{\mathcal{K}}(K_i)$ enaka pri vsakem $i = 1, \ldots, n$ in zato enaka $1/n$.

Pokažimo še obratno, da dana pogoja zagotavljata popolno tajnost. Pokazali smo že, da iz pogoja (2) v izreku sledi, da lahko za dan tajnopis y ključe in pripadajoče čistopise indeksiramo tako kot zgoraj ter da potem velja $P_{\mathcal{C}}(y|x_i) = P_{\mathcal{K}}(K_i)$. Po pogoju (1) v izreku je potem $P_{\mathcal{C}}(y|x_i) = 1/n$. Iz enačbe (1) sedaj določimo še $P_{\mathcal{C}}(y)$. Iz pogoja (2) v izreku in relacije $|\mathcal{C}| = |\mathcal{P}|$ dobimo, da za izbran $y \in \mathcal{C}$ množica $\{d_K(y)\,;\, y \in C(K)\}$ pokrije celo množico $\mathcal{P}$, kar pomeni, da je

$$P_{\mathcal{C}}(y) = \sum_{\{K\,;\, y \in C(K)\}} P_{\mathcal{K}}(K) P_{\mathcal{P}}(d_K(y)) = (1/n) \sum_{\{K\,;\, y \in C(K)\}} P_{\mathcal{P}}(d_K(y)) = 1/n\,.$$

S pomočjo Bayesovega obrazca sedaj dobimo

$$P_{\mathcal{P}}(x|y) = \frac{P_{\mathcal{P}}(x) P_{\mathcal{C}}(y|x)}{P_{\mathcal{C}}(y)} = \frac{P_{\mathcal{P}}(x)\; 1/n}{1/n} = P_{\mathcal{P}}(x)\,,$$

kar nam zagotavlja popolno tajnost. ■

Ideja Vigenèrejevega kriptosistema je tudi danes zelo pogosta pri konstrukciji kriptosistemov. Za uporabo v računalnikih črke zamenjamo z biti, seštevanje in odštevanje po modulu 2 pa je bolj poznano kot XOR seštevanje, ki ga označimo z $\oplus$. Če dan ključ uporabimo le enkrat, je to kriptosistem s popolno tajnostjo, poznan pa je pod imenom **enkratni ščit** (one-time pad). Gilbert Vernam je patentiral enkratni ščit že leta 1917, ko so ga začeli uporabljati pri šifriranju telegrafskih sporočil. Da zagotavlja popolno tajnost pa je dokazal Shannon šele 30 let kasneje. Vendar pa so vsi kriptosistemi, ki zagotavljajo popolno tajnost, zaradi dejstva, da mora biti $|\mathcal{K}| \geq |\mathcal{P}|$, neuporabni za komercialne namene, saj si je potrebno na varen način izmenjati veliko količino ključa. So pa seveda nepogrešljivi za vojaške in predvsem diplomatske namene, kjer zahtevamo, da napadalec Oskar ne sme nikoli ugotoviti, kaj je bila vsebina sporočila, ne glede na napredek v znanosti (rešitev težkih problemov v matematiki, razvoj algoritmov) in razvoju računalniške opreme. Za komercialne potrebe se ponavadi zadovoljimo z računsko varnostjo.

1.4 TOKOVNI TAJNOPISI IN GENERATORJI NAKLJUČNOSTI

V tem razdelku bomo podali aksiomatsko definicijo tokovnih tajnopisov ter njihov zelo pogost primer uporabe, ki spominja na enkratni ščit, to je šifriranje s pomočjo generatorja psevdonaključnih zaporedij bitov. Tak kriptosistem nam ne more zagotavljati brezpogojne varnosti, lahko pa zagotavlja računsko varnost. Le to bomo podali z definicijo kriptografsko varnega generatorja psevdonaključnih zaporedij bitov.

Tokovni tajnopis je sedmerica $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{L}, \mathcal{F}, \mathcal{E}, \mathcal{D})$ za katero velja:

1. $\mathcal{P}$ je končna množica možnih **čistopisov**,
2. $\mathcal{C}$ je končna množica možnih **tajnopisov**,
3. $\mathcal{K}$ je končna množica možnih **ključev**,
4. $\mathcal{L}$ je končna množica **tokovne abecede**,
5. $\mathcal{F} = (f_1, f_2, \dots)$ je **generator toka ključev**:

$$f_i : \mathcal{K} \times \mathcal{P}^{i-1} \to \mathcal{L} \qquad \text{za } i \geq 1\,,$$

6. za vsak tok ključev $z \in \mathcal{L}$ imamo natanko določen **šifrirni postopek** $e_z \in \mathcal{E}$ in ustrezen **dešifrirni postopek** $d_z \in \mathcal{D}$, kjer sta $e_z : \mathcal{P} \to \mathcal{C}$ in $d_z : \mathcal{C} \to \mathcal{P}$ taki funkciji, da je $d_z(e_z(x)) = x$ za vsak čistopis $x \in \mathcal{P}$.

V ozadju je torej postopek, ki nam na podlagi enega ključa K zgradi tok ključev $\boldsymbol{z} = z_1 \dots z_n$, s katerimi potem šifriramo 'tok' čistopisa $\boldsymbol{x} = x_1 \dots x_n$. S tem poskusimo zagotoviti vsaj računsko varnost tokovnega kriptosistema, saj vemo, da ni varno šifrirati več čistopisov z istim ključem.

Pogosto je tok ključev neodvisen od čistopisa, kar pomeni $f_i : \mathcal{K} \to \mathcal{L}$ za $i \geq 1$. Primer takega tokovnega tajnopisa je šifriranje z generatorjem psevdonaključnih bitov, kjer je $\mathcal{P} = \mathcal{C} = \mathcal{L} = \mathbb{Z}_2$ in $\mathcal{K} = (\mathbb{Z}_2)^m$. Generator je opisan z $\mathcal{F}$, njegova varnost pa je odvisna od izbire ključa $K \in \mathcal{K}$ in težavnosti funkcij f_i, ki naj bi bile **enosmerne** (one-way). Pravimo, da je funkcija $f : X \to Y$ enosmerna, če jo lahko opišemo z algoritmom polinomske časovne zahtevnosti parametra $|X|$, za njen inverz pa tak algoritem ni znan in se domneva, da ne obstaja (NP-polni problemi). Zaporedje psevdonaključnih bitov $\boldsymbol{z} = z_1 \dots z_n$ nam predstavlja tok ključev, s pomočjo katerega zašifriramo čistopis $\boldsymbol{x} = x_1 \dots x_n$ po pravilu

$$y_i = x_i \oplus z_i \tag{2}$$

in tako dobimo tok tajnopisa $\boldsymbol{y} = y_1 \dots y_n$. Vidimo torej, da je uporaba tokovnega tajnopisa, pravzaprav večkratna uporaba običajnega kriptosistema, kjer prikrivamo dejstvo, da vseskozi uporabljamo isti ključ. Kvaliteta prikrivanja pa nam predstavlja stopnjo računske varnosti.

Generator naključnih zaporedij bitov (**RBG** - random bit generator) je naprava ali algoritem, ki sestavlja zaporedja naključnih (kar pomeni $P(\texttt{bit} = \texttt{1}) = \frac{1}{2}$), in med seboj neodvisnih bitov. Taki generatorji so lahko narejeni s strojno ali pa s programsko opremo. Bolj varni so prvi (šumni generator, ki npr. izkorišča termično šumenje tranzistorja), saj je tako opazovalec onemogočen, da bi opazoval, oziroma vplival na naključnost.

Generator psevdonaključnih zaporedij bitov (**PRBG** - pseudorandom bit generator) je deterministični[4] algoritem, ki kot vhodne podatke dobi zaporedje bitov, generirano s pravim RBG, imenovano **seme**, dolžine k ter vrne zaporedje binarnih števil dolžine $l \gg k$, ki izgleda naključno. Kdaj zaporedje izgleda dovolj naključno, je težko opredeliti, saj je opredelitev odvisna tudi od namena uporabe generatorja. Lahko pa bi rekli, da takrat, ko ga statistični testi ne zavrnejo.

Izhodni podatki PRBG niso naključni, saj izmed vseh možnih zaporedij dolžine l (2^l jih je) lahko dobimo le en del teh, toliko kot je različnih semen (2^k). Seveda pa bi radi naredili tak PRBG, da nihče, ki bi opazoval izhodno zaporedje, ne bi mogel učinkovito ločiti to zaporedje od takega, ki bi ga dal pravi RBG. Da dosežemo zaupanje, da je PRBG res dober, mora ustrezati čimveč (vsem, ki jih naredimo) statističnim testom, ki preverjajo, če ima opazovani PRBG take lastnosti, kot naj bi jih imeli pravi RBG. Takih testov je lahko toliko, kot je lastnosti RBG, torej neskončno. Naredimo pa seveda tiste, ki bi nam lahko pomagali pri razbitju algoritma. Nekaj je preprostih standardnih, t.i. osnovnih testov (npr: ničel naj bi bilo približno toliko kot enk), nekaj pa bolj zapletenih. V četrtem poglavju so opisani testi iz programa Crypt-XS (osnovni testi, test linearnosti ter kompresijski test) in iz programa Diehard, ki jih je razvil George Marsaglia in so bolj zahtevni, saj iščejo najpogostejše šibke točke generatorjev, ki so povzročene zaradi neizogibne uporabe npr. linearnih komponent. Da testi ne pokažejo nobenih napak na PRBG, je potreben, ne pa tudi zadosten pogoj, da je PRBG dober in s tem tudi varen.

Pravimo, da PRBG **ustreza vsem polinomskim statističnim testom**, če noben algoritem polinomske časovne zahtevnosti ne more pravilno določiti razlike med zaporedjem generiranim s tem PRBG in zaporedjem generiranim z RBG z verjetnostjo večjo od 1/2.

Pravimo, da PRBG **ustreza napovednemu testu**, če noben algoritem polinomske časovne zahtevnosti ne more na podlagi danih prvih l bitov izhodnega zaporedja s določiti naslednji bit, to je bit s_{l+1}, zaporedja s z verjetnostjo večjo od 1/2.

Naslednji izrek nam pove, da sta definiciji ekvivalentni, dokaz pa si lahko preberemo v [18, str. 365–368]. Definiciji nam karakterizirata računsko varnost z enačbo (2) opisanega tokovnega tajnopisa, zato PRBG, ki ustreza eni od definicij, pravimo **kriptografsko varen PRBG**.

Izrek 1.4.1. *(**Univerzalnost napovednega testa.**) PRBG ustreza napovednemu testu, natanko tedaj ko ustreza vsem polinomskim statističnim testom.*

1.5 LFSR IN ZAHTEVA PO NELINEARNOSTI

Sestavni del mnogih PRBG je **linearni pomični register** (**LFSR** - linear feedback shift register), saj se ga da zelo dobro hardware-sko implementirati, izhodno zaporedje pa ima veliko periodo in zelo dobre statistične lastnosti. Vendar pa zaradi linearnih operacij, ki jih uporablja,

[4] Determinističen pomeni, da je izhodno zaporedje natanko določeno s semenom.

ni kriptografsko varen. V tem razdelku ga bomo opisali, prikazali bomo njegove dobre in slabe lastnosti ter povedali, kako lahko slednje odpravimo.

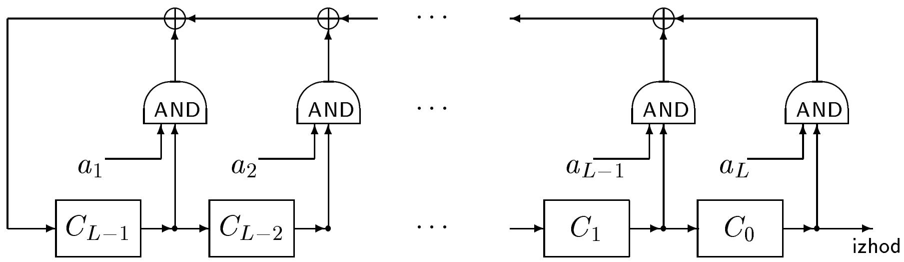
1.5.1 Lastnosti LFSR

Če je začetno stanje LFSR enako $[c_{L-1}, \dots, c_1, c_0]$, potem izhodno zaporedje registra določimo z rekurzivno formulo

$$c_j = (a_1 c_{j-1} + a_2 c_{j-2} + \dots + a_L c_{j-L}) \mod 2 \qquad \text{za } j \geq L\,.$$

Opišimo sedaj še dejansko izvedbo LFSR v strojni opremi. LFSR dolžine L vsebuje L celic $C_0, C_1, \dots, C_{L-1}$, vsaka pa lahko shrani en bit in ima en vhod in en izhod. Register vsebuje še uro, ki daje takt in s tem kontrolira gibanje podatkov. Ob vsakem taktu se izvedejo naslednje operacije:

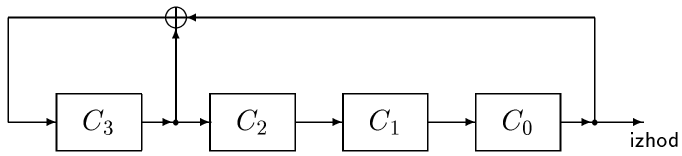
1. vsebina celice C_0 je izhod registra in tvori člen izhodnega zaporedja,
2. vsebina celice C_i se premakne v celico C_{i-1} za vsak i, $1 \leq i \leq L-1$, in
3. nova vsebina celice C_{L-1} je **povratni bit** b, ki ga izračunamo kot vsoto po modulu 2 (XOR) vsebine fiksne podmnožice celic registra (**povratne vezave**).



Slika 2: Delovanje LFSR

LFSR označimo z $\langle L, p(x)\rangle$, kjer je L dolžina LFSR, $p(x) = 1 + a_1 x + a_2 x^2 + \dots + a_L x^L$ pa **polinom povratnih vezav** , kjer je koeficient a_i enak 1, če je celica C_i celica povratne vezave, sicer pa enak 0. Slika 2 ponazarja shemo delovanja.

Primer 1.5.1. *Denimo, da imamo LFSR $\langle L, 1 + x + x^4\rangle$. Ponazorimo ga lahko s sliko*



Če je začetno stanje registra enako $[0,1,0,0]$, potem je izhodno zaporedje enako

$$s = \underbrace{0,0,1,0}_{seme},0,0,1,1,1,1,0,1,0,1,1,\underbrace{0,0,1,0}_{ponovitev},0,0,1,\dots$$

Pri računanju smo si pomagali z enačbo $s_j = s_{j-1} \oplus s_{j-4}$.

Ker je notranjih stanj registra končno mnogo, bomo pri pogoju, da je polinom povratnih povezav stopnje L, vedno dobili periodično izhodno zaporedje. V zgornjem primeru je perioda enaka 15. V primeru, da je ta polinom stopnje $L-k$, je izhodno zaporedje periodično, če izbrišemo prvih k bitov. V tem primeru so celice $C_1, \ldots, C_k$ brez efekta in jih lahko odstranimo. Od sedaj naprej bomo privzeli, da je vodilni koeficient polinoma povratnih povezav različen od nič, t.j.

$$a_L = 1\,.$$

Naj bo $p(x) \in \mathbb{Z}_2[x]$ polinom stopnje vsaj 1. Pravimo, da je polinom $p(x)$ **nerazcepen** nad $\mathbb{Z}_2$, če se ga ne da zapisati kot produkt dveh nekonstantnih polinomov. Npr. polinom $x^4 + x^2 + 1$ ni nerazcepen nad $\mathbb{Z}_2$, saj ga lahko zapišemo kot $x^4 + x^2 + 1 = (x^2 + x + 1)^2$. Za nerazcepen polinom $p(x)$ pravimo, da je **primitiven** nad $\mathbb{Z}_2$, če je polinom x generator grupe vseh neničelnih elementov faktorske grupe $\mathbb{Z}_2[x]/p(x)$. Če bralec še ni podkovan z znanjem o algebrskih strukturah, lahko najde osnovne definicije v [15, Chap. 2.5], vendar bomo bistvo razdelka poskusili predstaviti tako, da ga bo razumel tudi brez tega znanja.

V nadaljevanju bomo zapisali nekaj najpomembnejših dejstev o LFSR. Dokazali jih ne bomo, saj je naš osnovni namen pokazati, zakaj je LFSR tako uporaben in zakaj je potrebno testirati linearnost pri ugotavljanju kriptografske varnosti. Reference za dokaze pa bralec lahko sam najde v priročniku [15, str. 216, 4. odst.].

Trditev 1.5.2. *Če je $p(x)$ nerazcepen nad $\mathbb{Z}_2$, potem vsako od $2^L - 1$ neničelnih začetnih stanj nesingularnega LFSR $\langle L, p(x)\rangle$ dá izhodno zaporedje s periodo dolgo najmanjšemu številu N, takemu da $p(x)$ deli $1 + x^N$ v $\mathbb{Z}_2[x]$. (N bo vedno delitelj števila $2^L - 1$.) Če pa je $p(x)$ še primitiven, potem vsako od $2^L - 1$ neničelnih začetnih stanj nesingularnega LFSR $\langle L, p(x)\rangle$ dá izhodno zaporedje z največjo možno periodo, $N = 2^L - 1$.* ■

Od tu naprej bomo vedno predpostavljali, da je začetno stanje LFSR neničelno. V [15, Alg. 4.78] je opisan algoritem za naključno generiranje primitivnih polinomov nad $\mathbb{Z}_2$, vendar pa je za konstrukcijo LFSR, ki generira zaporedja z največjo možno periodo, dovolj pogledati v tabelo primitivnih polinomov [15, Table 4.8] za stopnje od 1 do 229. Če LFSR uporablja primitivni polinom povratnih povezav, potem mu pravimo **LFSR z največjo dolžino**, izhodnemu zaporedju pa **m-zaporedje**. To izhodno zaporedje ima zelo lepe statistične lastnosti, ki so opisane v naslednjih dveh trditvah.

Trditev 1.5.3. *Naj bo $k \in \mathbb{N}$, $1 \leq k \leq L$, in naj bo $\bar{s}$ katerokoli podzaporedje izhodnega zaporedja s in naj bo dolžine $2^L + k - 2$. Potem se vsako neničelno zaporedje dolžine k pojavi v zaporedju $\bar{s}$ natanko 2^{L-k}-krat, vsako ničelno zaporedje dolžine k pa natanko $(2^{L-k} - 1)$-krat.* ■

Z drugimi besedami, porazdelitev vzorcev fiksne dolžine, manjše od L, je v $\bar{s}$ skoraj enakomerna. Definirajmo še vzorce zaporedja, katerih dolžina je spremenljiva. Podzaporedje $\bar{s} = s_k, s_{k+1}, \ldots, s_{k+l-1}$ zaporedja $s = s_0, s_1, \ldots$, katerega vsak bit je enak 1, predhodni bit v zaporedju s, to je s_{k-1}, in naslednji bit, to je s_{k+l}, pa sta enaka 0, imenujemo **blok** (block); če pa je vsak bit podzaporedja $\bar{s}$ enak 0, predhodni in naslednji bit podzaporedja $\bar{s}$ v zaporedju s pa sta enaka 1, imenujemo $\bar{s}$ **vrzel** (gap). Vsako podzaporedje, ki je ali blok ali vrzel imenujemo **odsek** (run). Če npr. zaporedje 0010110100011111 razdelimo na posamezne odseke, dobimo

$$\underbrace{0,0}_{\text{vrzel}},\underbrace{1}_{\text{blok}},\underbrace{0}_{\text{vrzel}},\underbrace{11}_{\text{blok}},\underbrace{0}_{\text{vrzel}},\underbrace{1}_{\text{blok}},\underbrace{0,0,0}_{\text{vrzel}},\underbrace{1,1,1,1,1}_{\text{blok}}\,.$$

Izhodno zaporedje LFSR z največjo dolžino ustreza tudi **Golombovim postulatom naključnosti**, kar je potreben pogoj, da dano periodično binarno zaporedje izgleda naključno. Ti postulati so našteti v naslednji trditvi.

Trditev 1.5.4. *Naj bo s m-zaporedje s periodo $N = 2^L - 1$. Potem*
- *se število enk za največ 1 razlikuje od števila ničel,*
- *je vsaj 1/2 odsekov dolžine 1, vsaj 1/4 dolžine 2, vsaj 1/8 dolžine 3 ... vse dokler so v zaporedju odseki iskane dolžine ter*
- **avtokorelacijska funkcija**

$$C(t) = \frac{1}{N}\sum_{i=0}^{N-1}(2s_i - 1)(2s_{i+t} - 1)\,, \qquad \text{za } 0 \le t \le N-1$$

zavzame le dve vrednosti:

$$C(t) = \begin{cases} 1 & ;\ t = 0 \\ \frac{K}{N} & ;\ 1 \le t \le N-1 \end{cases},$$

kjer je K neko celo število. ■

Avtokorelacijska funkcija meri podobnost med istoležnimi členi zaporedja s in zaporedja, ki ga dobimo, če pomaknemo s za t mest v desno. Funkcija lahko zavzame vrednosti med -1 (vseh N parov bitov enakih $\{0,0\}$) in 1 (vseh N parov bitov enakih $\{1,1\}$)

1.5.2 Linearna zahtevnost

Pravimo, da LFSR $\langle L, p(x)\rangle$ **generira** binarno zaporedje s, če obstaja kakšno začetno stanje, s katerim dobimo izhodno zaporedje s in pravimo, da LFSR $\langle L, p(x)\rangle$ **generira** končno binarno zaporedje s^n dolžine n, če obstaja kakšno začetno stanje, s katerim dobimo izhodno zaporedje s, katerega začetni del je s^n. **Linearno zahtevnost** $L(s)$ poljubnega binarnega zaporedja s definiramo kot

$$L(s) = \begin{cases} 0 & ;\ s \text{ je ničelno zaporedje} \\ \infty & ;\ \text{noben LFSR ne generira } s \\ \text{dolžina najkrajšega LFSR, ki generira } s & ;\ \text{sicer} \end{cases}$$

Naštejmo nekaj osnovnih lastnosti linearne zahtevnosti.

Trditev 1.5.5. *Naj bosta s in t binarni zaporedji in s^n binarno zaporedje dolžine n. Potem velja:*
- $0 \le L(s^n) \le n$ *za vsak* $n \ge 1$,
- $L(s^n) = 0$ *natanko tedaj ko je* s^n *ničelno zaporedje,*
- $L(s^n) = n$ *natanko tedaj ko je* $s^n = 0, 0, \ldots, 0, 1$,
- $L(s) \le N$, *če je s periodično s periodo N in*
- $L(s \oplus t) \le L(s) + L(t)$. ■

Povejmo še en razlog, zakaj za LFSR uporabljamo nerazcepne polinome.

Trditev 1.5.6. *Če je polinom $p(x) \in \mathbb{Z}_2[x]$ stopnje L nerazcepen nad $\mathbb{Z}_2$, potem vsako od $2^L - 1$ neničelnih začetnih stanj nesingularnega LFSR $\langle L, p(x)\rangle$ dá izhodno zaporedje z linearno zahtevnostjo L.* ■

V naslednjih trditvah bomo povzeli, kolikšna naj bi bila linearna zahtevnost in kako naj bi naraščala pri zaporedjih generiranih s pravim RBG.

Trditev 1.5.7. *Naj bo zaporedje s^n naključno enakomerno izbrano iz množice vseh binarnih zaporedij dolžine n. Z B označimo parnostno funkcijo, ki je definirana kot $B(n) = n \bmod 2$. Potem sta matematično upanje in varianca za linearno zahtevnost zaporedja s^n enaka*

$$E[L(s^n)] = \frac{n}{2} + \frac{4+B(n)}{18} - \frac{1}{2^n}\left(\frac{n}{3}+\frac{2}{9}\right) \quad \textit{in}$$

$$\operatorname{var}[L(s^n)] = \frac{86}{81} - \frac{1}{2^n}\left(\frac{14-B(n)}{27}n + \frac{82-2B(n)}{81}\right) - \frac{1}{2^{2n}}\left(\frac{1}{9}n^2 + \frac{4}{27}n + \frac{4}{81}\right). \qquad \blacksquare$$

Naslednja trditev nam pove, da s testiranjem linearnosti hkrati testiramo tudi periodičnost.

Trditev 1.5.8. *Naj bo zaporedje s^n naključno izbrano iz množice vseh binarnih zaporedij dolžine n, kjer je $n = 2^t$ za nek $t \geq 1$, in naj bo s neskončno binarno zaporedje s periodo n, dobljeno s ponavljanjem zaporedja s^n. Potem je matematično upanje za linearno zahtevnost zaporedja s enako*

$$E[L(s)] = E[L(s^n)] = n - 1 + 2^{-n}. \qquad \blacksquare$$

Naj bo $s = s_0, s_1, \ldots$ binarno zaporedje in L_k linearna zahtevnost podzaporedja $s^k = s_0, s_1, \ldots, s_{k-1}$. Zaporedje $L_1, L_2, \ldots$ imenujemo **pregled linearne zahtevnosti** (linear complexity profile) zaporedja s. Analogno definiramo za zaporedja s^n dolžine n. V tem primeru je pregled linearne zahtevnosti zaporedja s^n enak $L_1, L_2, \ldots, L_n$. Pregled linearne zahtevnosti zaporedja dolžine n lahko izračunamo s pomočjo **Berlekamp-Masseyevega algoritma**, glej [15, Alg. 6.30], katerega časovna zahtevnost je le $\mathcal{O}(n^2)$. Algoritem nam za vsak k, $1 \leq k \leq n$, določi linearno zahtevnost L_k prvih k členov vhodnega zaporedja ter generator LFSR, ki generira to podzaporedje. Naštejmo nekaj osnovnih lastnosti pregleda linearne zahtevnosti, ki jih nazorno predstavlja Slika 3 k primeru 1.5.11.

Trditev 1.5.9. *Če je $L_1, L_2, \ldots$ pregled linearne zahtevnosti binarnega zaporedja $s = s_0, s_1, \ldots$, potem velja:*

1. *za vsak $j > i$ je $L_j \geq L_i$,*
2. *$L_{k+1} > L_k$ je možno le, če je $L_k \leq k/2$ in*
3. *če je $L_{k+1} > L_k$, potem je $L_{k+1} + L_k = k + 1$.* $\blacksquare$

Poglejmo sedaj še, kakšen je pregled linearne zahtevnosti za naključno izbrana zaporedja.

Trditev 1.5.10. *Naj bo $s = s_0, s_1, \ldots$ zaporedje, zgenerirano s pravim RBG. Z s^k označimo podzaporedje $s_0, s_1, \ldots, s_{k-1}$ zaporedja s, z d_k pa najmanjše število j, za katerega je $L_{k+j} > L_k$. Iz druge točke prejšnje trditve vidimo, da lahko d_k definiramo le za tiste k, kjer je $L_k \leq k/2$. V tem primeru je matematično upanje za $E(d_k)$ enako*

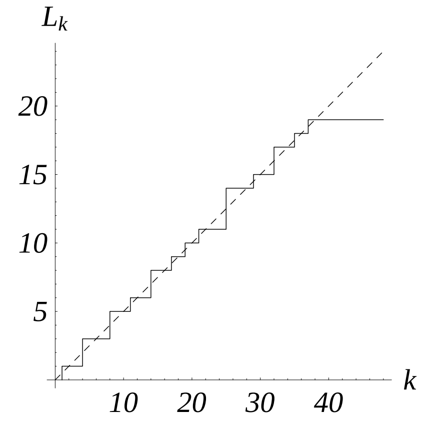
$$E(d_k) = 2. \qquad \blacksquare$$

Kako iz teh lastnosti zaporedij, generiranih z RBG, naredimo statistični test za preverjanje naključnosti, je opisano v 7. testu Crypt-XS testov na strani 56. Naslednji primer predstavlja pregled linearne zahtevnosti za periodično zaporedje, Slika 3 pa pomaga k razumevanju prejšnjih dveh trditev.

Primer 1.5.11. *Naj bo s periodično zaporedje s periodo 20 in ciklom*

$$s^{20} = 1,0,0,1,0,0,1,1,1,1,0,0,0,1,0,0,1,1,1,0.$$

Pregled linearne zahtevnosti je potem enak $1,1,1,3,3,3,3,5,5,5,6,6,6,8,8,8,9,9,10,10,11,11,$-$11,11,14,14,14,14,15,15,15,17,17,17,18,18,19,19,\ldots,19,\ldots$.



Slika 3: Graf pregleda linearne zahtevnosti zaporedja iz primera 1.5.11

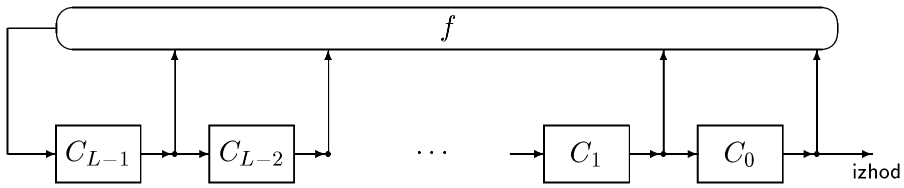
Iz naslednjih dveh trditev sledi, da LFSR ni kriptografsko varen PRBG.

Trditev 1.5.12. *Naj bo s^n končno binarno zaporedje dolžine n, njegova linearna zahtevnost pa naj bo L. Potem obstaja enoličen LFSR dolžine L, ki generira s^n, natanko tedaj ko je $L \leq n/2$.* ■

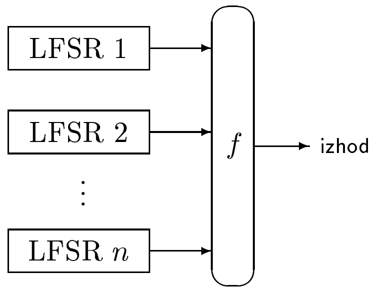
Trditev 1.5.13. *Naj bo s (neskončno) binarno zaporedje z linearno zahtevnostjo L in naj bo t (lahko končno in ne nujno začetni del) podzaporedje zaporedja s dolžine vsaj $2L$. Potem nam Berlekamp-Masseyev algoritem z vhodnim zaporedjem t določi LFSR dolžine L, ki generira s.* ■

1.5.3 Nelinearizacija LFSR

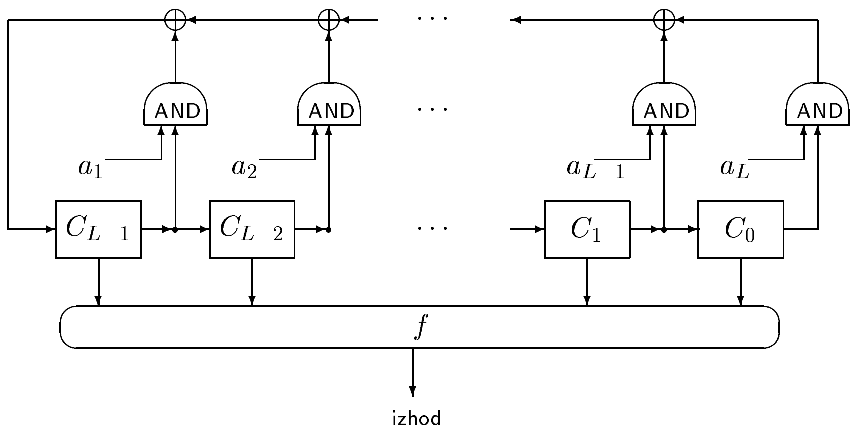
Ker nam linearnost LFSR povzroči, da LFSR ni več kriptografsko varen PRBG, bi radi naredili podoben PRBG, tako da bi ohranili dobre osnovne statistične lastnosti izhodnega zaporedja, linearnost pa bi odpravili. Na Sliki 4, Sliki 5 in Sliki 6 so prikazani trije splošni koncepti, kako uporabljati nelinearno funkcijo $f : (\mathbb{Z}_2)^n \to \mathbb{Z}_2$, v [15, Chap. 6] pa so opisani tudi konkretni primeri in podane reference za napade na nekatere od njih.



Slika 4: Nelinearni pomični register (nonlinear feedback shift register)



Slika 5: Nelinearno združevanje LFSR generatorjev



Slika 6: LFSR z nelinearnim filtrom

2 OSNOVE VERJETNOSTNEGA RAČUNA

Za podrobno razumevanje ozadja statističnih testov je potrebno znanje verjetnostnega računa, katerega osnove dobimo v [9] ali pa v [6], ter znanje statistike, ki ga bomo podali v naslednjem poglavju, kaj več pa najdemo v [8] ali pa v [5]. V našteti literaturi je snov podrobno predstavljena, jaz pa bom izpostavil le tisti del celotne teorije, ki ga bomo potrebovali za splošen opis statističnih testov (3. pogl.) ter opis statističnih testov za PRBG (4. pogl.).

Prvi trije razdelki opisujejo slučajne spremenljivke in njihove lastnosti. Navedenih je nekaj najbolj znanih diskretnih ter zveznih porazdelitev. Zadnji trije razdelki pa so namenjeni konvergenci zaporedja slučajnih spremenljivk. Konvergenco navadno ugotavljamo s pomočjo karakterističnih funkcij, za preproste primere pa lahko uporabimo kar centralni limitni izrek (2.4.1) ali pa njegovo posplošitev, izrek (2.6.6), ki ga lahko uporabimo za statistiko pri večrazsežni normalni porazdelitvi.

2.1 PORAZDELITVE SLUČAJNIH SPREMENLJIVK

Vsakemu izidu iz verjetnostnega prostora lahko pripišemo neko vrednost. Tako pri metu kocke pogledamo, koliko pik je na zgornji ploskvi. Število pik je diskretna slučajna spremenljivka. Lahko pa si naključno izberemo dva avtomobila na parkirišču in izmerimo razdaljo med njima. Ta razdalja je sedaj zvezna slučajna spremenljivka. V tem razdelku bomo podali bolj aksiomatski opis zgoraj naštetih pojmov.

Verjetnostni prostor je trojica $(\Omega, \mathcal{F}, P)$, kjer je Ω neprazna **množica izidov**, $\mathcal{F}$ σ-algebra[5] nad Ω ter P **verjetnostna mera** nad **družino dogodkov** $\mathcal{F}$. Pravimo, da je $\mathcal{F} \subset \mathcal{P}(X)$ σ-algebra nad množico Ω, če velja:

1. $\Omega \in \mathcal{F}$,
2. če je $A \in \mathcal{F}$, je tudi $A^C \in \mathcal{F}$ in
3. če so $A_1, A_2, \ldots$ elementi družine $\mathcal{F}$, je tudi $\displaystyle\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$.

Verjetnostna mera nad σ-algebro $\mathcal{F}$ pa je funkcija $P : \mathcal{F} \to [0,1]$ pri kateri za poljubne paroma disjunktne množice $A_1, A_2, \ldots$ iz $\mathcal{F}$ velja

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)\,.$$

Za funkcijo $X : \Omega \to \mathbb{R}$ rečemo, da je **slučajna spremenljivka**, če je $\{\omega \in \Omega; X(\omega) \leq x\} \in \mathcal{F}$ za vsak $x \in \mathbb{R}$. Verjetnost $P(\{\omega \in \Omega; X(\omega) \leq x\})$ bomo ponavadi označili kar s $P(X \leq x)$. Slučajno spremenljivko X lahko opišemo tudi s **porazdelitveno funkcijo**

$$F_X(x) = P(X \leq x)\,,$$

ki jo enolično določa.

Za slučajno spremenljivko X pravimo, da je **diskretna**, če slika v kakšno števno veliko podmnožico realnih števil. V tem primeru jo lahko opišemo tudi z verjetnostno shemo

$$X \sim \begin{pmatrix} x_1 & x_2 & x_3 & \ldots \\ p_1 & p_2 & p_3 & \ldots \end{pmatrix},$$

kjer so v zgornji vrsti naštete vrednosti iz zaloge vrednosti, v spodnji pa verjetnosti $p_i = P(X = x_i)$, s katerimi zavzame te vrednosti. Znak $\sim$ uporabljamo, ko opisujemo porazdelitev, lahko z

[5] Pojem se uporablja pri Lebesgueovem integralu, ki je opisan v [16], ki je učbenik za teorijo mere.

verjetnostno shemo, lahko z neko znano porazdelitvijo, npr. $X \sim \mathrm{Geom}(p)$, lahko pa rečemo, da je X enako porazdeljena kot Y, kar označimo z $X \sim Y$. Njeno **matematično upanje** (expectation) je pričakovana aritmetična sredina – povprečje – in je podano z

$$E(X) = \sum_{n=1}^{\infty} x_n \, p_n \,,$$

varianca, ki pa je mera za pričakovano odstopanje od povprečja, pa z

$$\mathrm{var}(X) = E[(X - E(X))^2] \,.$$

Če za slučajno spremenljivko X obstaja integrabilna funkcija f_X, ki določa porazdelitveno funkcijo na način

$$F_X(x) = \int_{-\infty}^{x} f_X(t) dt \,,$$

potem pravimo, da je X **zvezna** slučajna spremenljivka z **gostoto** f_X. Verjetnost, da X zavzame vrednost v neki **Borelovi množici**[6] A, je torej enaka

$$P(X \in A) = \int_A f_X(t) dt \,.$$

Tudi zvezni slučajni spremenljivki X lahko priredimo **matematično upanje** in **varianco**:

$$E(X) = \int_{-\infty}^{\infty} x f_X(x) dx \,, \qquad \mathrm{var}(X) = E[(X - E(X))^2] \,.$$

Ker je matematično upanje v obeh primerih linearen funkcional, je varianca enaka

$$\mathrm{var}(X) = E(X^2) - E(X)^2 \,.$$

V primeru zvezne porazdelitve izračunamo $E(X^2)$ kot

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x) dx \,.$$

Seveda ni odveč opomba, da o matematičnem upanju in varianci lahko govorimo le, če navedene vsote in integrali sploh obstajajo.

V statističnih testih bomo uporabljali nekaj najbolj znanih porazdelitev slučajnih spremenljivk, ki jih bomo definirali v nadaljevanju. Za vsako bomo navedli oznako in porazdelitev. Matematičnega upanja in variance ne bomo računali, ampak ju bomo le zapisali. Diskretne slučajne spremenljivke so najpogosteje celoštevilske.

1. Metanje kovanca lahko opišemo z **Bernoulijevo slučajno spremenljivko** s parametrom p (pri poštenem kovancu je $p = \frac{1}{2}$), kjer $X = 1$ pomeni, da je padel grb, $X = 0$ pa, da je padla cifra:

$$X \sim \begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix} \,.$$

[6] Pri Lebesgueovem integralu na $\mathbb{R}$ uporabljamo **Borelovo σ-algebro** nad $\mathbb{R}$. Po definiciji je to najmanjša σ-algebra, ki vsebuje vse odprte množice, njene elemente pa imenujemo **Borelove množice**. Tudi zaprte množice so potem Borelove.

2. Če vržemo n kovancev, je število grbov X **binomska slučajna spremenljivka**:

$$X \sim \mathrm{Bin}(n,p)\,, \quad P(X=k) = \binom{n}{k} p^k (1-p)^{n-k} \qquad \text{za } 0 \le k \le n\,,$$
$$E(X) = np\,, \quad \mathrm{var}(X) = np(1-p)\,.$$

3. Če mečemo kovanec toliko časa, da pade grb, in z X označimo število potrebnih metov, vključno z zadnjim, je X porazdeljena po zakonu **geometrijske slučajne spremenljivke**:

$$X \sim \mathrm{Geom}(p)\,, \quad P(X=k) = (1-p)^{k-1} p \qquad \text{za } k \ge 1\,,$$
$$E(X) = \frac{1}{p}\,, \quad \mathrm{var}(X) = \frac{(1-p)}{p^2}\,.$$

4. Če z X označimo število metov, vključno z zadnjim, do m-tega grba, potem dobimo **negativno binomsko slučajno spremenljivko**:

$$X \sim \mathrm{NegBin}(m,p)\,, \quad P(X=k) = \binom{k-1}{m-1} p^m (1-p)^{k-m} \qquad \text{za } k \ge m\,,$$
$$E(X) = \frac{m}{p}\,, \quad \mathrm{var}(X) = \frac{m(1-p)}{p^2}\,.$$

5. Zelo znana je tudi **Poissonova porazdelitev**, za katero velja:

$$X \sim \mathrm{Po}(\lambda)\,, \quad P(X=k) = \frac{\lambda^k}{k!} e^{-\lambda}\,,$$
$$E(X) = \lambda\,, \quad \mathrm{var}(X) = \lambda\,.$$

Pri zveznih slučajnih spremenljivkah pa so najpogostejše naslednje porazdelitve, ki jih pogosto, predvsem pri statističnih testih, dobimo kot limite diskretnih porazdelitev.

6. **Enakomerna porazdelitev**:

$$X \sim I(0,1)\,, \quad f_X(x) = \begin{cases} 1 \ ; \ 0 < x < 1 \\ 0 \ ; \ \text{sicer} \end{cases}\,,$$
$$E(X) = \frac{1}{2}\,, \quad \mathrm{var}(X) = \frac{1}{12}\,.$$

7. **Normalna porazdelitev**:

$$X \sim N(\mu, \sigma^2)\,, \quad f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}\,,$$
$$E(X) = \mu\,, \quad \mathrm{var}(X) = \sigma^2\,.$$

Spremenljivko X, porazdeljeno po zakonu $\sim N(0,1)$, imenujemo **standardna normalna slučajna spremenljivka**.

8. **Gama porazdelitev**[7]:

$$X \sim \Gamma(\rho, \lambda)\,, \quad f_X(x) = \begin{cases} \dfrac{\lambda^\rho}{\Gamma(\rho)} x^{\rho-1} e^{-\lambda x} & ; \ x > 0 \\ 0 & ; \ \text{sicer} \end{cases} \qquad \text{za } \rho > 0 \text{ in } \lambda > 0\,,$$
$$E(X) = \frac{\rho}{\lambda}\,, \quad \mathrm{var}(X) = \frac{\rho}{\lambda^2}\,.$$

[7] Opisana je s pomočjo **Eulerjeve funkcije** Γ, definirane na intervalu $(0, \infty)$ s predpisom $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$.

9. χ^2**-porazdelitev** :

$$X \sim \chi^2(\nu)\,,\quad f_X(x) = \begin{cases} \dfrac{1}{\Gamma(\nu/2)\,2^{\nu/2}}\, x^{\nu/2-1} e^{-x/2} & ; \ x \ge 0 \\ 0 & ; \ \text{sicer} \end{cases} \qquad \text{za } \nu > 0\,,$$

$$E(X) = \nu\,,\quad \mathrm{var}(X) = 2\nu\,.$$

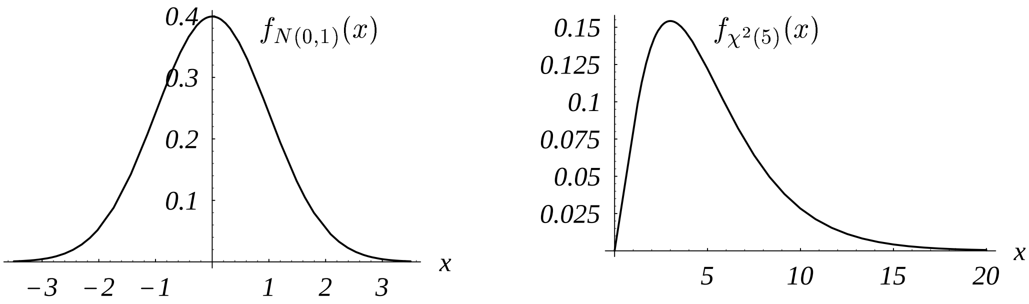
Parameter ν imenujemo **število prostostnih stopenj**. Opazimo tudi $\chi^2(\nu) \sim \Gamma(\nu/2, 1/2)$.

10. **Eksponentna porazdelitev**:

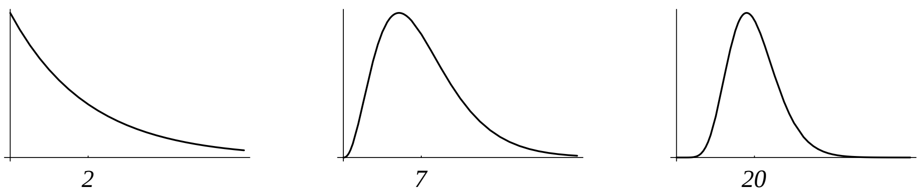
$$X \sim \exp(\lambda)\,,\quad f_X(x) = \begin{cases} \lambda e^{-\lambda x} & ; \ x \ge 0 \\ 0 & ; \ \text{sicer} \end{cases} \qquad \text{za } \lambda > 0\,,$$

$$E(X) = \frac{1}{\lambda}\,,\quad \mathrm{var}(X) = \frac{1}{\lambda^2}\,.$$

Tudi eksponentna porazdelitev je poseben primer gama porazdelitve, velja namreč $\exp(\lambda) \sim \Gamma(1, \lambda)$.



Slika 7: Grafa gostot slučajnih spremenljivk, porazdeljenih po zakonu $N(0,1)$ in $\chi^2(5)$



Slika 8: Grafi gostot slučajnih spremenljivk, porazdeljenih po zakonu $\chi^2(2)$, $\chi^2(7)$ in $\chi^2(20)$

Graf gostote standardne normalne porazdelitve in graf gostote $\chi^2(5)$ porazdelitve sta prikazana na Sliki 7. Na Sliki 8 pa je prikazano, da je oblika gostote za $\chi^2(\nu)$ porazdelitev za $\nu \le 2$ bistveno drugačna kot za $\nu > 2$. Slika 8 ponazarja tudi, da se ta oblika za velike ν približuje tisti za normalno porazdelitev $N(\nu, 2\nu)$.

2.2 NEODVISNOST SLUČAJNIH SPREMENLJIVK

Podobno kot slučajne spremenljivke definiramo tudi slučajne vektorje. Funkcija

$$\boldsymbol{X} : \Omega \to \mathbb{R}^n$$

je **slučajni vektor**, če je $\{\omega \in \Omega; \boldsymbol{X}(\omega) \leq \boldsymbol{x}\} \in \mathcal{F}$ za vsak $\boldsymbol{x} \in \mathbb{R}^n$. Zapis $\boldsymbol{x} \leq \boldsymbol{y}$ pomeni, da je $x_i \leq y_i$ za vsak i, $1 \leq i \leq n$, kjer so x_i in y_i komponente vektorjev $\boldsymbol{x}$ in $\boldsymbol{y}$. Tudi slučajni vektor $\boldsymbol{X}$ opišemo s **porazdelitveno funkcijo**

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = P(\boldsymbol{X} \leq \boldsymbol{x}),$$

ki ga enolično določa. Ni se težko prepričati, glej npr. [9, str. 83], da je $\boldsymbol{X}$ slučajni vektor, natanko tedaj ko so njegove komponente X_i slučajne spremenljivke.

Za slučajni vektor $\boldsymbol{X}$ pravimo, da je **zvezno porazdeljen**, če obstaja **gostota**, torej funkcija, ki določa porazdelitveno funkcijo na način

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \int_{-\infty}^{\boldsymbol{x}} f_{\boldsymbol{X}}(\boldsymbol{t}) \boldsymbol{dt}.$$

Integral $\int_{-\infty}^{\boldsymbol{x}}$ pomeni, da integriramo po množici $\{\boldsymbol{t}; \boldsymbol{t} \leq \boldsymbol{x}\}$.

Za slučajne spremenljivke $X_1, \ldots, X_n$ rečemo, da so **neodvisne**, če je

$$P(X_1 \in A_1, \ldots, X_n \in A_n) = P(X_1 \in A_1) \cdot \ldots \cdot P(X_n \in A_n)$$

za poljubne Borelove množice $A_1, \ldots, A_n$, kar pomeni, da je vrednost, ki jo zavzame spremenljivka X_i, povsem neodvisna od vrednosti, ki jih zavzamejo ostale spremenljivke. Analogno definiramo neodvisnost tudi za slučajne vektorje.

Neodvisnost dveh slučajnih spremenljivk X in Y merimo tudi s **kovarianco**, ki je definirana z

$$\operatorname{cov}(X, Y) = E[(X - E(X))(Y - E(Y))].$$

Enačbo poenostavimo v

$$\operatorname{cov}(X, Y) = E(XY) - E(X)E(Y).$$

Če sta X in Y neodvisni, se da pokazati, glej [9, str. 133], da je

$$E(XY) = E(X)E(Y). \tag{3}$$

To pomeni, da je $\operatorname{cov}(X, Y) = 0$, če sta X in Y neodvisni. Obrat v splošnem ne velja. Če imamo več slučajnih spremenljivk, jih zložimo v vektor, medsebojne odvisnosti pa shranimo v **kovariančno matriko** tega vektorja, ki je definirana kot

$$[\operatorname{var}(\boldsymbol{X})]_{ij} = \operatorname{cov}(X_i, X_j).$$

Če sta $\boldsymbol{X}$ in $\boldsymbol{Y}$ neodvisna slučajna vektorja, je $\operatorname{cov}(X_i, Y_j) = 0$ za vsak par i, j, kar pomeni

$$\operatorname{var}(\boldsymbol{X} + \boldsymbol{Y}) = \operatorname{var}(\boldsymbol{X}) + \operatorname{var}(\boldsymbol{Y}).$$

Če je $\boldsymbol{X}$ konstanten vektor, je $\operatorname{var}(\boldsymbol{X}) = 0$.

Iz Fubinijevega izreka[8] dobimo, glej [6, str. 99], da če ima slučajni vektor $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ gostoto, jo imajo tudi vse njegove komponente,

$$f_{X_1}(x_1) = \int_{\mathbb{R}^{n-1}} f_{\boldsymbol{X}}(x_1, \ldots, x_n) dx_2 \ldots dx_n.$$

[8] Fubinijev izrek je opisan v [16] in nam pove, kdaj smemo zamenjati vrstni red integriranja pri večkratnih integralih.

V tem primeru so $X_1, \dots, X_n$ neodvisne, natanko tedaj ko je gostota vektorja enaka produktu gostot njegovih komponent

$$f_{\boldsymbol{X}} = f_{X_1} \cdot \dots \cdot f_{X_n} \,. \tag{4}$$

Tudi slučajnemu vektorju $\boldsymbol{X}$ lahko določimo **matematično upanje** $E(\boldsymbol{X})$ z:

$$E(\boldsymbol{X}) = [E(X_1), \dots, E(X_n)]^T \,.$$

Trditev 2.2.1. *Če je slučajni vektor $\boldsymbol{X}$ zvezno porazdeljen, potem je*

$$E(\boldsymbol{X}) = \int_{\mathbb{R}^n} \boldsymbol{t} f_{\boldsymbol{X}}(\boldsymbol{t}) \boldsymbol{dt} \,.$$

Dokaz.

$$\begin{aligned} E(\boldsymbol{X}) &= \int_{\mathbb{R}^n} \boldsymbol{t} f_{\boldsymbol{X}}(\boldsymbol{t}) \boldsymbol{dt} = \left[\int_{\mathbb{R}^n} t_i f_{\boldsymbol{X}}(\boldsymbol{t}) \boldsymbol{dt} \right]_{i=1}^n = \\ &= \left[\int_{\mathbb{R}} t_i \int_{\mathbb{R}^{n-1}} f_{\boldsymbol{X}}(\boldsymbol{t}) \boldsymbol{dt} \right]_{i=1}^n = \left[\int_{\mathbb{R}} t_i f_{X_i}(t_i) dt_i \right]_{i=1}^n = [E(X_i)]_{i=1}^n \,. \end{aligned}$$

■

Iz linearnosti matematičnega upanja za slučajne spremenljivke sledi, da je matematično upanje linearno tudi v več dimenzijah. Torej, če je $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ in $\boldsymbol{b} \in \mathbb{R}^m$, je

$$E(\boldsymbol{AX} + \boldsymbol{b}) = \boldsymbol{A} E(\boldsymbol{X}) + \boldsymbol{b} \,.$$

2.3 TRANSFORMACIJE SLUČAJNIH SPREMENLJIVK

Pogosto nas zanima, kako iz porazdelitve spremenljivk X in Y določimo porazdelitev spremenljivk kot sta npr. $X + Y$ ali pa $\log X$. V pomoč nam je naslednji izrek.

Izrek 2.3.1. *Naj bo $\boldsymbol{X}$ slučajni vektor z vrednostmi v $\mathbb{R}^n$ in U odprta množica, za katero velja $P(\boldsymbol{X} \in U) = 1$. Naj bo V odprta množica v $\mathbb{R}^n$ in $\Phi : U \to V$ zvezno odvedljiva bijektivna preslikava z zvezno odvedljivim inverzom Ψ. Če je $f_{\boldsymbol{X}}$ gostota slučajnega vektorja $\boldsymbol{X}$, potem je gostota slučajnega vektorja $\boldsymbol{Y} = \Phi(\boldsymbol{X})$ enaka*

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = f_{\boldsymbol{X}}(\Psi(\boldsymbol{y})) \cdot |\det(J_\Psi(\boldsymbol{y}))| \,,$$

kjer je J_Ψ Jacobijeva matrika preslikave Ψ.

Dokaz. Ker je Φ bijektivna, je za poljubno Borelovo množico $B \subset V$

$$\boldsymbol{Y} \in B \Leftrightarrow \boldsymbol{X} \in \Psi(B) \,,$$

torej $P(\boldsymbol{Y} \in B) = P(\boldsymbol{X} \in \Psi(B))$, ali zapisano z gostotami

$$\int_B f_{\boldsymbol{Y}}(\boldsymbol{y}) \boldsymbol{dy} = \int_{\Psi(B)} f_{\boldsymbol{X}}(\boldsymbol{x}) \boldsymbol{dx} \,.$$

V integral na desni strani enačbe uvedemo novo spremenljivko $\boldsymbol{y} = \Phi(\boldsymbol{x})$ in dobimo

$$\int_B f_{\boldsymbol{Y}}(\boldsymbol{y}) \boldsymbol{dy} = \int_B f_{\boldsymbol{X}}(\Psi(\boldsymbol{y})) \cdot |\det(J_\Psi(\boldsymbol{y}))| \boldsymbol{dy} \,.$$

Ker to velja za vsako Borelovo množico B, sta funkciji znotraj integrala enaki. ■

Primer 2.3.2. *Če sta $X \sim \Gamma(a,\lambda)$ in $Y \sim \Gamma(b,\lambda)$ neodvisni slučajni spremenljivki, kako je potem porazdeljena slučajna spremenljivka $X+Y$? Preslikavo $\Phi : (0,\infty)\times(0,\infty) \to (0,1)\times(0,\infty)$ definiramo kot*

$$\Phi(X,Y) = (\frac{X}{X+Y}, X+Y)\,.$$

Na tak način zato, da bosta komponenti neodvisni. Določimo njen inverz

$$\Psi(U,V) = (UV, V-UV)$$

ter Jacobijevo matriko

$$J_\Psi(u,v) = \begin{bmatrix} v & u \\ -v & 1-u \end{bmatrix}.$$

Uporabimo izrek (2.3.1):

$$f_{U,V}(u,v) = f_{X,Y}(uv, v-uv)\cdot v\,.$$

Ker sta X in Y neodvisni, je "gostota enaka produktu gostot"

$$\begin{aligned} f_{U,V}(u,v) &= f_X(uv)f_Y(v-uv)v = \\ &= \frac{\lambda^a}{\Gamma(a)}(uv)^{a-1}e^{-\lambda uv}1_{(0,\infty)}(uv)\frac{\lambda^b}{\Gamma(b)}(v-uv)^{b-1}e^{-\lambda(v-uv)}1_{(0,\infty)}(v-uv)v = \\ &= \frac{\lambda^{a+b}}{\Gamma(a)\Gamma(b)}u^{a-1}(1-u)^{b-1}v^{a+b-1}e^{-\lambda v}1_{(0,1)}(u)1_{(0,\infty)}(v) = \\ &= \underbrace{\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}u^{a-1}(1-u)^{b-1}1_{(0,1)}(u)}_{\text{funkcija spremenljivke } u}\,\underbrace{\frac{\lambda^{a+b}}{\Gamma(a+b)}v^{a+b-1}e^{-\lambda v}1_{(0,\infty)}(v)}_{\text{funkcija spremenljivke } v} \end{aligned}$$

Ker je gostota enaka produktu dveh funkcij ene spremenljivke, sta slučajni spremenljivki U in V neodvisni in gostota spremenljivke V, torej $X+Y$, je enaka

$$f_V = \frac{\lambda^{a+b}}{\Gamma(a+b)}v^{a+b-1}e^{-\lambda v}1_{(0,\infty)}(v)\,,$$

torej je $X+Y \sim \Gamma(a+b,\lambda)$.

Če sta $X \sim \chi^2(a)$ in $Y \sim \chi^2(b)$ neodvisni slučajni spremenljivki, je slučajna spremenljivka $X+Y$ porazdeljena po zakonu $\chi^2(a+b)$, kar lahko preverimo, tako da v zadnjem primeru vzamemo $\lambda = \frac{1}{2}$.

Podobno lahko izračunamo gostote še za mnogo drugih slučajnih spremenljivk in njihovih transformacij, zato bomo tiste, ki jih bomo potrebovali kasneje, tu le našteli. Naj bodo $X, Y,$- $X_1, \ldots, X_n$ v vseh primerih neodvisne. Veljajo naslednje relacije:

$$X \sim \Gamma(a,\lambda) \text{ in } Y \sim \Gamma(b,\lambda) \Longrightarrow X+Y \sim \Gamma(a+b,\lambda)\,, \tag{5}$$

$$X \sim \chi^2(a) \text{ in } Y \sim \chi^2(b) \Longrightarrow X+Y \sim \chi^2(a+b)\,, \tag{6}$$

$$X_1 \sim \chi^2(1), \ldots, X_n \sim \chi^2(1) \Longrightarrow X_1 + \cdots + X_n \sim \chi^2(n)\,, \tag{7}$$

$$X \sim N(\mu,\sigma) \text{ in } a \in \mathbb{R} \Longrightarrow aX \sim N(a\mu, (a\sigma)^2)\,, \tag{8}$$

$$X \sim N(\mu,\sigma) \text{ in } Y \sim N(\nu,\rho) \Longrightarrow X+Y \sim N(\mu+\nu, \sigma^2+\rho^2)\,, \tag{9}$$

$$X \sim N(\mu, \sigma^2) \Longrightarrow \frac{X-\mu}{\sigma} \sim N(0,1) \text{ in} \tag{10}$$

$$X \sim N(0,1) \Longrightarrow X^2 \sim \chi^2(1)\,. \tag{11}$$

Zvezno slučajno spremenljivko lahko preoblikujemo, tako da ima enakomerno porazdelitev. To bomo storili takrat, ko nas bo zanimala verjetnost, da je vrednost slučajne spremenljivke manjša od izbrane.

Trditev 2.3.3. *Naj bo X zvezna slučajna spremenljivka z gostoto f_X. Če definiramo*

$$Y = \int_{-\infty}^{X} f_X(x)dx\,,$$

potem je Y enakomerno porazdeljena slučajna spremenljivka.

Dokaz. Izračunajmo porazdelitveno funkcijo spremenljivke $Y = F_X(X)$. Pri poljubnem $y \in (0,1)$ je

$$F_Y(y) = P(Y \le y) = P(F_X(X) \le y)\,.$$

Naj bo $x_0 = \sup_x\{F_X(x) \le y\}$. Ker je vsaka porazdelitvena funkcija naraščajoča in ker je v našem primeru tudi zvezna, je $F_X(X) \le y$, natanko tedaj ko je $X \le x_0$, kar pomeni

$$F_Y(y) = P(X \le x_0) = F_X(x_0)\,.$$

Ker vsaka zvezna porazdelitvena funkcija doseže vse vrednosti na (0,1), je tudi vrednost y dosežena. Dosežena je pri x_0, ker smo ga tako izbrali. Torej je

$$F_Y(y) = y\,,$$

to pa je porazdelitvena funkcija enakomerno porazdeljene slučajne spremenljivke. ■

2.4 CENTRALNI LIMITNI IZREK

V statistiki so zanimivi primeri, ko se porazdelitev zaporedja diskretnih slučajnih spremenljivk približuje porazdelitvi neke zvezne slučajne spremenljivke. Kako dobra je aproksimacija z limitno porazdelitvijo, je odvisno od tega, kateri člen zaporedja vzamemo, pri statističnih raziskavah to pomeni, kako velik vzorec vzamemo.

Ker obravnavamo diskretne in zvezne slučajne spremenljivke, bomo za primerjavo morali izbrati količino, ki jo lahko določimo obema. Izbrali bomo porazdelitveno funkcijo. Pravimo, da zaporedje slučajnih spremenljivk $X_1, X_2, \ldots$ **konvergira v porazdelitvi** (distribution) k slučajni spremenljivki X, če za vsako točko x, kjer je F_X zvezna, velja

$$\lim_{n\to\infty} F_{X_n}(x) = F_X(x)\,.$$

Tako konvergenco opišemo z relacijo

$$X_n \xrightarrow{d} X \quad \text{ali pa} \quad X = \overset{d}{\lim_{n\to\infty}} X_n\,.$$

Analogno definiramo konvergenco v porazdelitvi za slučajne vektorje.

Izrek 2.4.1. (**Centralni limitni izrek.**) *Naj bodo slučajne spremenljivke* $X_1, \ldots, X_n$ *neodvisne in enako porazdeljene z* $E(X_1) = \mu$ *in* $\mathrm{var}(X_1) = \sigma^2 < \infty$. *Za* $S_n = X_1 + \cdots + X_n$ *velja*

$$\frac{S_n - n\mu}{\sqrt{n\sigma^2}} \xrightarrow{d} N(0,1).$$ ■

Dobro intuitivno skico dokaza najdemo v [6], celoten dokaz pa je v [10]. Izrek pove, da se za dovolj velike n porazdelitev S_n približuje porazdelitvi $N(n\mu, n\sigma^2)$. Oceno hitrosti konvergence nam da Berry-Esseenov izrek, ki pa ga tu ne bomo navedli, glej npr. [4].

2.5 KARAKTERISTIČNE FUNKCIJE

Pri večrazsežni normalni porazdelitvi bomo potrebovali izrek, da je porazdelitvena funkcija $F_{\boldsymbol{X}}$ slučajnega vektorja $\boldsymbol{X}$ natanko določena s svojo karakteristično funkcijo. Tu bomo naredili le povzetek teorije o karakterističnih funkcijah, ki pa je podrobno opisana v [9] ali [6].

Naj bo $\boldsymbol{X}$ n-razsežni slučajni vektor. Funkcijo $\phi_{\boldsymbol{X}} : \mathbb{R}^n \to \mathbb{R}$, definirano s predpisom

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = E(e^{i\boldsymbol{t}^T\boldsymbol{X}}),$$

imenujemo **karakteristična funkcija** slučajnega vektorja $\boldsymbol{X}$.

Trditev 2.5.1. *Karakteristična funkcija obstaja pri vsaki porazdelitvi.* ■

Izrek 2.5.2. (**Izrek o edinosti.**) *Porazdelitvena funkcija* $F_{\boldsymbol{X}}$ *slučajnega vektorja* $\boldsymbol{X}$ *je natanko določena s svojo karakteristično funkcijo* $\phi_{\boldsymbol{X}}$. ■

To pomeni, da je vseeno ali porazdelitev slučajnega vektorja $\boldsymbol{X}$ opišemo s porazdelitveno funkcijo $F_{\boldsymbol{X}}$ ali pa z njegovo karakteristično funkcijo $\phi_{\boldsymbol{X}}$.

Centralni limitni izrek lahko dokažemo s pomočjo karakterističnih funkcij. Uporabimo izrek o konvergenci karakterističnih funkcij. Konkretno pa ga bomo uporabili pri dokazu izreka (3.3.1), ki govori o limitni porazdelitvi Pearsonove χ^2-statistike.

Izrek 2.5.3. *Naj bodo* $\boldsymbol{X}_n$ *slučajni vektorji s karakterističnimi funkcijami* $\phi_{\boldsymbol{X}_n}$. *Če funkcije* $\phi_{\boldsymbol{X}_n}$ *konvergirajo po točkah proti* g *in je* g *zvezna v* $\boldsymbol{0}$, *potem je* g *karakteristična funkcija nekega slučajnega vektorja* $\boldsymbol{X}$ *in velja*

$$\boldsymbol{X}_n \xrightarrow{d} \boldsymbol{X}.$$ ■

Ni se težko prepričati, glej npr. [9, str. 164], da je karakteristična funkcija standardno normalno porazdeljene slučajne spremenljivke X enaka

$$\phi_X(t) = e^{-\frac{1}{2}t^2}. \tag{12}$$

2.6 VEČRAZSEŽNA NORMALNA PORAZDELITEV

Naj bodo $X_1, \ldots, X_n$ normalno porazdeljene slučajne spremenljivke, ki so med seboj lahko odvisne in določajo večrazsežno normalno porazdelitev. Cilj tega razdelka je, da določimo porazdelitev njihovega odstopanja od pričakovanih vrednosti. Spremenljivke bomo zapisali kot komponente slučajnega vektorja $\boldsymbol{X}$ z matematičnim upanjem $\boldsymbol{\mu}$ in s kovariančno matriko $\boldsymbol{\Sigma}$. Poiskali bomo rang r matrike $\boldsymbol{\Sigma}$ in njen posplošen inverz $\boldsymbol{\Sigma}^-$. Nato bomo uporabili izrek (2.6.6), ki je dokazan v tem razdelku in pravi, da je slučajna spremenljivka $(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^-(\boldsymbol{X}-\boldsymbol{\mu})$

porazdeljena po zakonu $\chi^2(r)$. Rezultate razdelka bomo potrebovali za obrazložitev 2. in 16. Diehard testa, ki sta opisana v razdelku 4.4, razdelek kot celoto pa za dokaz izreka (3.3.1), ki govori o limitni porazdelitvi Pearsonove χ^2-statistike.

Naj bodo $Z_1, \ldots, Z_n$ neodvisne standardno normalno porazdeljene slučajne spremenljivke. Zložimo jih v vektor $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^T$. Za poljubno matriko $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ in vektor $\boldsymbol{\mu} \in \mathbb{R}^m$ pravimo, da je slučajni vektor

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{\mu}$$

večrazsežno normalno porazdeljen. Ker je vsaka komponenta vektorja $\boldsymbol{X}$,

$$X_i = \left(\sum_{j=1}^{n} a_{ij} Z_j \right) + b_i \qquad i = 1, \ldots, m\,,$$

afina kombinacija neodvisnih standardno normalno porazdeljenih slučajnih spremenljivk in zato normalno porazdeljena slučajna spremenljivka, je $\boldsymbol{X}$ slučajni vektor.

Trditev 2.6.1. *Karakteristična funkcija večrazsežno normalno porazdeljenega slučajnega vektorja $\boldsymbol{X}$, ki ga dobimo iz zgoraj definirane matrike $\boldsymbol{A}$ in vektorja $\boldsymbol{Z}$, je enaka*

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} e^{-\frac{1}{2}\boldsymbol{t}^T\boldsymbol{\Sigma}\boldsymbol{t}}\,,$$

kjer je

$$\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T\,.$$

Dokaz. Za zgoraj definirani vektor $\boldsymbol{X}$ izračunamo karakteristično funkcijo:

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = \phi_{\boldsymbol{A}\boldsymbol{Z}+\boldsymbol{\mu}}(\boldsymbol{t}) = E(e^{i\boldsymbol{t}^T(\boldsymbol{A}\boldsymbol{Z}+\boldsymbol{\mu})}) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} E(e^{i\boldsymbol{t}^T\boldsymbol{A}\boldsymbol{Z}})\,.$$

Vpeljemo nov vektor $\boldsymbol{s} = \boldsymbol{A}^T\boldsymbol{t}$ in dobimo

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} E(e^{i\boldsymbol{s}^T\boldsymbol{Z}}) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} E\left(e^{i\sum_{j=1}^{n} s_j Z_j}\right) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} E\left(\prod_{j=1}^{n} e^{is_j Z_j}\right)\,,$$

oziroma če upoštevamo še, da so spremenljivke Z_i neodvisne in s tem tudi spremenljivke $e^{is_j Z_j}$, potem iz enačbe (3) sledi

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} \prod_{j=1}^{n} E(e^{is_j Z_j})\,.$$

Matematično upanje $E(e^{is_j Z_j})$ je pravzaprav karakteristična funkcija standardno normalno porazdeljene slučajne spremenljivke, ki je zapisana v enačbi (12), zato je

$$\phi_{\boldsymbol{X}}(\boldsymbol{t}) = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} \prod_{j=1}^{n} e^{-\frac{1}{2}s_j^2} = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} e^{-\frac{1}{2}\boldsymbol{s}^T\boldsymbol{s}} = e^{i\boldsymbol{t}^T\boldsymbol{\mu}} e^{-\frac{1}{2}\boldsymbol{t}^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{t}}\,. \qquad \blacksquare$$

Naslednja trditev pove, da je vsaka komponenta X_i slučajnega vektorja $\boldsymbol{X}$ porazdeljena po zakonu $N(\mu_i, (\boldsymbol{\Sigma})_{ii})$ in da je $\mathrm{cov}(X_i, X_j) = (\boldsymbol{\Sigma})_{ij}$.

Trditev 2.6.2. *Za zgoraj definirani večrazsežno normalno porazdeljen slučajni vektor $\boldsymbol{X}$ sta matematično upanje in kovariančna matrika enaka*

$$E(\boldsymbol{X}) = \boldsymbol{\mu} \qquad \textit{in} \qquad \mathrm{var}(\boldsymbol{X}) = \boldsymbol{\Sigma}\,,$$

kjer je $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T$.

Dokaz. Ker je matematično upanje linearno in ga računamo po komponentah, je

$$E(\boldsymbol{X}) = E(\boldsymbol{A}\boldsymbol{Z} + \boldsymbol{\mu}) = \boldsymbol{A}E(\boldsymbol{Z}) + \boldsymbol{\mu} = \boldsymbol{A}\boldsymbol{0} + \boldsymbol{\mu} = \boldsymbol{\mu}\,.$$

Izračunamo še kovariančno matriko $\boldsymbol{C} = [c_{ij}]_{i,j=1}^{m}$ vektorja $\boldsymbol{X}$:

$$\begin{aligned}
c_{ij} &= E(X_i X_j) - E(X_i)E(X_j) = \\
&= E\left[\left(\sum_{k=1}^{n} a_{ik}Z_k\right)\cdot\left(\sum_{l=1}^{n} a_{jl}Z_l\right)\right] - E\left(\sum_{k=1}^{n} a_{ik}Z_k\right)E\left(\sum_{l=1}^{n} a_{jl}Z_l\right) = \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} a_{ik}a_{jl}E(Z_kZ_l) - \sum_{k=1}^{n}\sum_{l=1}^{n} a_{ik}a_{jl}E(Z_k)E(Z_l) = \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} a_{ik}a_{jl}\big(E(Z_kZ_l) - E(Z_k)E(Z_l)\big) = \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} a_{ik}a_{jl}\operatorname{cov}(Z_k, Z_l)\,.
\end{aligned}$$

Ker so $Z_1, \ldots, Z_n$ neodvisne, je

$$\operatorname{cov}(Z_k, Z_l) = \begin{cases} 0 & ;\ k \neq l \\ \operatorname{var}(Z_k) & ;\ k = l \end{cases}.$$

Vzeli pa smo $\operatorname{var}(Z_k) = 1$, zato je

$$c_{ij} = \sum_{k=1}^{n} a_{ik}a_{jk} = (\boldsymbol{A}\boldsymbol{A}^T)_{ij} = (\boldsymbol{\Sigma})_{ij}\,,$$

kar smo želeli dokazati. ■

Iz zadnjih dveh trditev vidimo, da je karakteristična funkcija večrazsežno porazdeljenega normalnega vektorja odvisna le od njegovega matematičnega upanja in kovariančne matrike. Ker karakteristična funkcija enolično določa porazdelitev, je smiselno za parametra pri oznaki te porazdelitve uporabiti matematično upanje in kovariančno matriko. Oznaka je potem posplošitev tiste za eno dimenzijo, v njej navedemo le še razsežnost porazdelitve,

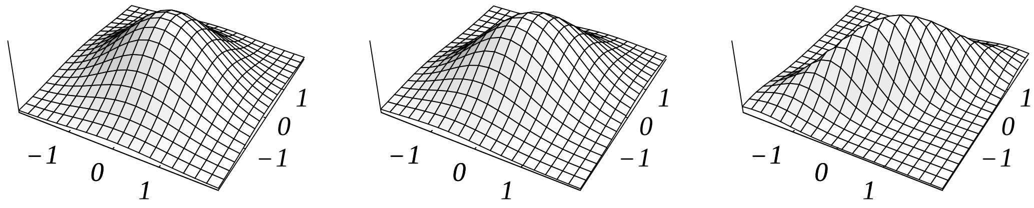
$$\boldsymbol{X} \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})\,.$$

Zanimiv je primer, ko je $n = m$ in ima matrika $\boldsymbol{A}$ poln rang, saj potem lahko s pomočjo izreka (2.3.1) izračunamo gostoto vektorja $\boldsymbol{X}$. Najprej izračunajmo gostoto vektorja $\boldsymbol{Z}$:

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \prod_{i=1}^{n} f_{Z_i}(z_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z_i^2} = \frac{1}{(2\pi)^{n/2}}e^{-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{I}\boldsymbol{z}}\,.$$

Ker so Z_i neodvisne, smo lahko uporabili enačbo (4) s strani 26, ki pravi da je potem gostota slučajnega vektorja enaka produktu gostot posameznih komponent. Če bi enačbo (4) uporabili na primeru $\boldsymbol{A} = \operatorname{diag}(d_1, \ldots, d_n)$, kjer so vsi $d_i \neq 0$, bi ugotovili, da so slučajne spremenljivke $X_1, \ldots, X_n$ neodvisne in porazdeljene $X_i \sim N(\mu_i, d_i^2)$, natanko tedaj ko je slučajni vektor $\boldsymbol{X}$ porazdeljen po zakonu $N_n(\boldsymbol{\mu}, \boldsymbol{D})$, kjer je $\boldsymbol{D} = \operatorname{diag}(d_1^2, \ldots, d_n^2)$. Ker je $\boldsymbol{A}$ obrnljiva, je Jacobijeva matrika preslikave $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{\mu}$ enaka kar $\boldsymbol{A}$, inverz preslikave pa preslikava $\boldsymbol{Z} = \boldsymbol{A}^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$. Zato lahko s pomočjo izreka (2.3.1) določimo gostoto vektorja $\boldsymbol{X}$:

$$\begin{aligned}
f_{\boldsymbol{X}}(\boldsymbol{x}) &= \frac{1}{|\det \boldsymbol{A}|}f_{\boldsymbol{Z}}(\boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})) = \frac{1}{(2\pi)^{n/2}|\det \boldsymbol{A}|}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{A}^{-T}\boldsymbol{I}(\boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\
&= \frac{1}{(2\pi)^{n/2}|\det \boldsymbol{A}|}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} = \frac{1}{(2\pi)^{n/2}\sqrt{\det \boldsymbol{\Sigma}}}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}(\boldsymbol{x}-\boldsymbol{\mu})}\,.
\end{aligned}$$



Slika 9: Grafi gostot dvorazsežno normalno porazdeljenih slučajnih vektorjev.

Na Sliki 9 so prikazani grafi gostot

$$N\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right),\ N\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1&\frac{1}{3}\\\frac{1}{3}&1\end{bmatrix}\right)\ \text{in}\ N\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1&\frac{4}{5}\\\frac{4}{5}&1\end{bmatrix}\right)$$

porazdeljenih slučajnih vektorjev. Vidimo, da se porazdelitev koncentrira na premici, ko se izvendiagonalni element po absolutni vrednosti približuje 1. Ta element je kovarianca med komponentama. V [9] je pokazano, da za poljubni slučajni spremenljivki X in Y, za kateri obstajata matematično upanje in varianca, velja

$$|\operatorname{cov}(X,Y)| \leq \sqrt{\operatorname{var}(X)\operatorname{var}(Y)}$$

in enakost je dosežena, natanko tedaj ko med X in Y obstaja linearna zveza. Če naš primer posplošimo na več dimenzij, to pomeni, da med komponentami na začetku razdelka definiranega večrazsežno normalno porazdeljenega slučajnega vektorja $\boldsymbol{X}$ obstaja linearna zveza, natanko tedaj ko je matrika $\boldsymbol{A}$, in zato tudi matrika Σ, singularna.

Primer 2.6.3. *Če sta slučajni spremenljivki X in Y normalno porazdeljeni, pokažimo da to še ni dovolj, da dobimo večrazsežno (lahko tudi izrojeno, npr. enorazsežno) normalno porazdelitev. Vzemimo $X \sim N(0,1)$ in*

$$Y = \begin{cases} X & ; -0,67 < X < 0,67 \\ -X & ; \text{sicer} \end{cases}.$$

Očitno je tudi slučajna spremenljivka Y standardno normalno porazdeljena, toda preslikava s katero smo dobili vektor $(X,Y)^T$ očitno ni afina, kar je razvidno tudi iz zaloge vrednosti slučajnega vektorja $(X,Y)^T$, ki je sestavljena iz treh komponent za povezanost. Naj nas podatek, da je kovariančna matrika pozitivno definitna (saj je $\operatorname{cov}(X,Y) \doteq 0'14$) ne zavede, ker je v tem primeru brez pomena.

Vrnimo se k splošni definiciji večrazsežne normalne porazdelitve. Naslednja trditev nam pove, da je slučajni vektor, dobljen z afino transformacijo normalno porazdeljenega slučajnega vektorja, tudi normalno porazdeljen.

Trditev 2.6.4. *Če je slučajni vektor $\boldsymbol{X}$ porazdeljen po zakonu $N_m(\boldsymbol{\mu},\boldsymbol{\Sigma})$ in $\boldsymbol{B} \in \mathbb{R}^{m\times p}$ obrnljiva matrika in $\boldsymbol{\nu} \in \mathbb{R}^p$, potem je porazdelitev slučajnega vektorja $\boldsymbol{BX}+\boldsymbol{\nu}$ enaka $N_p(\boldsymbol{B\mu}+\boldsymbol{\nu}, \boldsymbol{B\Sigma B}^T)$.*

Dokaz. Ker je $\boldsymbol{X}$ porazdeljen $N_m(\boldsymbol{\mu},\boldsymbol{\Sigma})$, je dobljen z afino transformacijo nekega standardno normalno porazdeljenega slučajnega vektorja $\boldsymbol{Z}$, torej je

$$\boldsymbol{X} = \boldsymbol{AZ} + \boldsymbol{\mu},$$

kjer je $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ in $\boldsymbol{\mu} \in \mathbb{R}^m$. Sedaj je

$$\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{X} + \boldsymbol{\nu} = \boldsymbol{B}(\boldsymbol{A}\boldsymbol{Z} + \boldsymbol{\mu}) + \boldsymbol{\nu} = \boldsymbol{B}\boldsymbol{A}\boldsymbol{Z} + \boldsymbol{B}\boldsymbol{\mu} + \boldsymbol{\nu}\,,$$

kar pomeni, da je slučajni vektor $\boldsymbol{Y}$ porazdeljen po zakonu $N_p(\boldsymbol{B}\boldsymbol{\mu}+\boldsymbol{\nu}, \boldsymbol{B}\boldsymbol{A}(\boldsymbol{B}\boldsymbol{A})^T) \sim N_p(\boldsymbol{B}\boldsymbol{\mu}+\boldsymbol{\nu}, \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T)$. ■

Matriki $\boldsymbol{A}$, za katero velja $\boldsymbol{A}^2 = \boldsymbol{A}$, pravimo **idempotentna matrika**. Naslednjo trditev potrebujemo za dokaz izreka (2.6.6).

Trditev 2.6.5. *Naj bo $\boldsymbol{A}$ simetrična idempotentna matrika velikosti $n \times n$ in slučajni vektor $\boldsymbol{Z}$ porazdeljen $N_n(\boldsymbol{0}, \boldsymbol{I})$. Potem je slučajna spremenljivka $\boldsymbol{Z}^T\boldsymbol{A}\boldsymbol{Z}$ porazdeljena po zakonu $\chi^2(r)$, kjer je $r = \mathrm{rang}(\boldsymbol{A})$.*

Dokaz. Ker je $\boldsymbol{A}$ simetrična, jo lahko diagonaliziramo, tako da je $\boldsymbol{A} = \boldsymbol{Q}^T\boldsymbol{\Lambda}\boldsymbol{Q}$, kjer je $\boldsymbol{\Lambda}$ diagonalna, $\boldsymbol{Q}$ pa ortogonalna matrika, $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$. In ker je $\boldsymbol{A}$ idempotentna matrika, lahko za $\boldsymbol{\Lambda}$ vzamemo

$$\boldsymbol{\Lambda} = \mathrm{diag}\{\underbrace{1,\ldots,1}_{r},0,\ldots,0\}\,.$$

Definirajmo slučajni vektor $\boldsymbol{W} = \boldsymbol{Q}\boldsymbol{Z}$, ki je porazdeljen $N_n(\boldsymbol{Q}\boldsymbol{0}, \boldsymbol{Q}\boldsymbol{Q}^T) \sim N_n(\boldsymbol{0}, \boldsymbol{I})$ in izračunajmo vrednost slučajne spremenljivke $\boldsymbol{Z}^T\boldsymbol{A}\boldsymbol{Z}$:

$$\boldsymbol{Z}^T\boldsymbol{A}\boldsymbol{Z} = \boldsymbol{Z}^T\boldsymbol{Q}^T\boldsymbol{\Lambda}\boldsymbol{Q}\boldsymbol{Z} = \boldsymbol{W}^T\boldsymbol{\Lambda}\boldsymbol{W} = W_1^2 + \cdots W_r^2\,.$$

Ker so W_i neodvisne in standardno normalno porazdeljene, saj so komponente $N_n(\boldsymbol{0}, \boldsymbol{I})$ porazdeljenega slučajnega vektorja $\boldsymbol{W}$, je ta porazdelitev enaka $\chi^2(r)$, kar smo dobili iz relacij (7) in (11) s strani 27. ■

Matrika $\boldsymbol{\Sigma}^-$ je posplošen inverz matrike $\boldsymbol{\Sigma}$, če velja $\boldsymbol{\Sigma}^-\boldsymbol{\Sigma}\boldsymbol{\Sigma}^- = \boldsymbol{\Sigma}^-$. Numerično ga lahko dobimo s pomočjo SVD[9] razcepa matrike $\boldsymbol{\Sigma}$.

Izrek 2.6.6. *Če je slučajni vektor $\boldsymbol{X}$ porazdeljen $N_n(\boldsymbol{0}, \boldsymbol{\Sigma})$, kjer je $r = \mathrm{rang}(\boldsymbol{\Sigma})$. Potem je slučajna spremenljivka $\boldsymbol{X}^T\boldsymbol{\Sigma}^-\boldsymbol{X}$ porazdeljena $\chi^2(r)$.*

Dokaz. Ker je matrika $\boldsymbol{\Sigma}$ simetrična nenegativno definitna, obstaja njen simetrični koren $\boldsymbol{A}$, torej je $\boldsymbol{\Sigma} = \boldsymbol{A}^2$ in $\boldsymbol{A} = \boldsymbol{A}^T$. Če vzamemo $\boldsymbol{Z} \sim N_n(\boldsymbol{0}, \boldsymbol{I})$, je $\boldsymbol{A}\boldsymbol{Z} \sim N_n(\boldsymbol{A}\boldsymbol{0}, \boldsymbol{A}\boldsymbol{A}) \sim \boldsymbol{X}$. Izpeljemo iskano relacijo

$$\boldsymbol{X}^T\boldsymbol{\Sigma}^-\boldsymbol{X} \sim \boldsymbol{Z}^T\boldsymbol{A}\boldsymbol{\Sigma}^-\boldsymbol{A}\boldsymbol{Z}\,.$$

Ker je $\boldsymbol{A}\boldsymbol{\Sigma}^-\boldsymbol{A}$ idempotentna matrika z rangom r, je izrek s pomočjo prejšnje trditve dokazan. ■

Če izrek uporabimo za slučajni vektor $\boldsymbol{X} - \boldsymbol{\mu}$, kjer je $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, je slučajna spremenljivka $(\boldsymbol{X} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^-(\boldsymbol{X} - \boldsymbol{\mu})$ porazdeljena po zakonu $\chi^2(r)$. V praksi bi izrek (2.6.6) lahko uporabili na sledeč način. Predpostavimo, da imamo n slučajnih spremenljivk X_i, kjer je $1 \leq i \leq n$. Tvorimo slučajni vektor $\boldsymbol{X} = (X_1, \ldots, X_n)$, izračunamo njegovo karakteristično funkcijo in če ugotovimo, da je večrazsežno normalno porazdeljen, potem za slučajni vektor $\boldsymbol{X} - \boldsymbol{\mu}$, kjer je $\boldsymbol{\mu}$ matematično upanje vektorja $\boldsymbol{X}$, lahko uporabimo izrek (2.6.6). Na tak način sta skonstruirana 2. in 16. Diehard test, ki sta opisana v razdelku 4.4.

[9] SVD je kratica za angleško ime Singular Value Decomposition. Postopek je opisan v:
J. W. DEMMEL, *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 1997, str. 109, str. 237.

3 PREVERJANJE STATISTIČNIH DOMNEV

V tem poglavju se bomo ukvarjali z matematično statistiko. Ta opisuje, kako lahko lastnost vzorcev neke populacije preverjamo s statističnimi testi ter kakšne so zaželjene lastnosti statističnih testov. Domneve o porazdelitvi izhodnega zaporedja PRBG so, kot bomo videli, neparametrične, zato bomo v tretjem razdelku navedli najbolj znana neparametrična statistična testa, to sta χ^2-test in test Kolmogorova. Matematično pravilnost χ^2-testa bomo tudi izpeljali. Neparametrični pomeni, da so namenjeni preverjanju oblike porazdelitve, parametrični testi pa so namenjeni ugotavljanju parametrov porazdelitve, katere obliko že privzamemo. Katerega od testov izberemo pri obravnavi slučajnega vzorca, je odvisno od vrste domneve, ki jo preverja, velikosti vzorca, pa tudi lastnosti testov (hitrost, doslednost ipd). Primerjava med χ^2-testom in testom Kolmogorova testom je narejena na koncu zadnjega razdelka. Če pa vzorec ni slučajen, kar pomeni, da ima med seboj odvisne elemente, potem lahko naredimo test s pomočjo izreka (2.6.6). Kako te teste uporabimo za testiranje generatorjev psevdonaključnih zaporedij bitov (PRBG), je opisano v naslednjem poglavju. Pravzaprav sta opisana dva programa, Crypt-XS in Diehard, ki vsak zase preverjata različne lastnosti PRBG.

Če se bralec prvič spopada s področjem matematične statistike, bo lažje razumel, kam se odvija rdeča nit tega poglavja, če si bo najprej prebral nazoren primer χ^2-testa na strani 44. Pri branju se bo mogoče vprašal, zakaj zavrnemo hipotezo o porazdelitvi ravno za tiste vzorce, kjer je vrednost Pearsonove χ^2-statistike zelo velika. Intuitiven odgovor bi bil: "Zato ker je to na nek način vzorec z ekstremno vrednostjo lastnosti, ki jo testiramo". Odgovor ni zadovoljiv, saj so majhne vrednosti tudi ekstremne. Da bomo razumeli način izbire intervala zaupanja, bomo v drugem razdelku vpeljali definiciji napak prve in druge vrste, interval zaupanja pa bomo vzeli tak, da bosta ti dve napaki čimmanjši.

3.1 OSNOVE MATEMATIČNE STATISTIKE

Ko preverjamo lastnosti neke serije izdelkov, je včasih nemogoče preveriti vse izdelke ali pa je to nesmotrno (recimo pri preizkusu lomljivosti porcelanastih krožnikov). Zato si izberemo neko naključno izbrano končno podmnožico vseh izdelkov in na njih preverimo dano lastnost. To je tipična situacija, s kakršnimi se ukvarja matematična statistika. Prevedemo jo v tako obliko, v kateri lahko uporabimo znanje verjetnostnega računa. Prostor izidov imenujemo **populacija**. Privzemimo, da lahko lastnost, ki jo preverjamo, predstavimo kot slučajno spremenljivko na populaciji. Primerno izbran del populacije, na kateri preverjamo lastnost, se imenuje **vzorec**. Da lahko z veliko verjetnostjo sklepamo iz vzorca na celo populacijo, mora biti ta vzorec slučajno izbran. To pomeni, da je bil vsak element v vzorcu naključno izbran iz celotne populacije, nekateri elementi se v vzorcu zato lahko tudi ponovijo. Tako izbran vzorec imenujemo **slučajen vzorec**. Pravzaprav nas ne zanima, kateri elementi $(\omega_1, \dots, \omega_n)$ so v vzorcu, ampak kakšna je vrednost slučajne spremenljivke X na njih. Če označimo

$$x_i = X(\omega_i) \qquad i = 1, \dots, n\,,$$

pravimo, da je $\boldsymbol{z} = (x_1, \dots, x_n)$ vzorec za spremenljivko X. Če je vzorec slučajen, je vektor $\boldsymbol{z}$ pravzaprav realizacija slučajnega vektorja

$$\boldsymbol{Z} = (X_1, \dots, X_n)\,,$$

katerega komponente so neodvisne slučajne spremenljivke in porazdeljene tako kot X.

Označimo s $k(x)$ število tistih elementov ω iz vzorca, pri katerih je $X(\omega) \leq x$. Funkcijo

$$V_n(x) = \frac{k(x)}{n}$$

imenujemo **vzorčna porazdelitvena funkcija** ali tudi **empirična porazdelitvena funkcija**. Zanjo velja, da skoraj gotovo konvergira k porazdelitveni funkciji slučajne spremenljivke X, kar nam pove Glivenkov izrek, poznan tudi kot **osnovni izrek matematične statistike**. Dokaz je zgolj preverjanje definicij, zato ga izpustimo, bralec pa ga najde v [8, str. 12].

Izrek 3.1.1. *(**Glivenkov izrek.**) Naj bo $\boldsymbol{Z} = (X_1, \ldots, X_n)$ slučajen vzorec za spremenljivko X. Če definiramo slučajne spremenljivke*

$$D_n = \sup_{x \in \mathbb{R}} |V_n(x) - F_X(x)| \,,$$

potem zaporedje $D_1, D_2, \ldots$ skoraj gotovo konvergira proti 0, torej

$$P(\lim_{n \to \infty} D_n = 0) = 1 \,. \qquad \blacksquare$$

Informacije o spremenljivki X v vzorcu Z navadno ne izkoristimo neposredno, ampak preko neke funkcije $U : \mathbb{R}^n \to \mathbb{R}^m$. Ponavadi je $m = 1$, saj si je lažje predstavljati podatek v eni kot pa v mnogo dimenzijah. Vedno bomo vzeli Borelovo funkcijo in jo imenovali **vzorčna statistika** ali na kratko kar **statistika**, včasih tudi **testna funkcija**, in jo označili z

$$U = U(X_1, \ldots, X_n) \,.$$

Omenimo dve najbolj znani statistiki. Prva je

$$U(X_1, \ldots, X_n) = \frac{1}{n} \sum_{k=1}^{n} X_k$$

in jo označimo z $\bar{X}$ ter poimenujemo **vzorčno povprečje**. Če je vzorec slučajen, je zaradi linearnosti matematičnega upanja

$$E(\bar{X}) = E(X) \qquad \text{in} \qquad \mathrm{var}(\bar{X}) = \frac{\mathrm{var}(X)}{n} \,.$$

Druga zelo znana statistika pa je $U(X_1, \ldots, X_n) = \frac{1}{n} \sum_{k=1}^{n} (X_k - \bar{X})^2$, ki ji pravimo **vzorčna varianca** in jo označimo z $V_{\boldsymbol{Z}}$.

3.2 PRESKUŠANJE HIPOTEZ

Vsaka domneva o neznani porazdelitveni funkciji F_X slučajne spremenljivke X, ki deluje na populaciji, se imenuje **statistična hipoteza**. Vse hipoteze, ki pridejo v poštev, imenujemo **dopustne**. Tisti, ki jo dejansko želimo preskusiti, pravimo **ničelna hipoteza**, vsem ostalim dopustnim pa **alternativne hipoteze**. Če z $\mathcal{D}$ označimo družino vseh dopustnih porazdelitev slučajne spremenljivke X in z $\mathcal{D}_0$ neprazno pravo podmnožico, v kateri domnevamo, da je F_X, potem našo hipotezo simbolično predstavimo kot

$$H(F_X \in \mathcal{D}_0)$$

ali na kratko H_0. Če ima množica $\mathcal{D}_0$ en sam element, potem pravimo, da je hipoteza $H(F_X \in \mathcal{D}_0)$ **enostavna**, sicer je **sestavljena**. Če je tip porazdelitvene funkcije znan in se hipoteza nanaša le na vrednosti parametrov, se imenuje **parametrična hipoteza**, vsaka drugačna hipoteza se imenuje **neparametrična hipoteza**.

Primer 3.2.1. *Za H_0 vzemimo hipotezo: "Slučajna spremenljivka X je porazdeljena normalno po zakonu $N(0,1)$." Ker se hipoteza ne nanaša le na vrednosti parametrov, ampak pravi tudi " ... je porazdeljena normalno ... ", je to neparametrična hipoteza. In ker je $\mathcal{D}_0 = \{N(0,1)\}$, je hipoteza H_0 enostavna.*

Parametrične hipoteze pa se v praksi najpogosteje uporabljajo pri merjenju fizikalnih količin, kar nam ponazarja naslednji primer.

Primer 3.2.2. *V pekarni pečejo 80 dag težke štruce kruha, toda vsi vemo, da so nekatere malce težje, druge pa malce lažje. Z X označimo težo slučajno izbrane štruce in pričakujemo $X \sim N(80,\sigma^2)$. Parameter σ se imenuje* **standardni odklon** *in ga ocenimo iz vzorca s pomočjo neke t.i.* **parametrične metode***.*

Ne glede na to, kakšen vzorec bomo dobili iz PRBG, bomo zanj domnevali, da je slučajen, kar pomeni, da je vrednost vsakega bita Bernoulijeva slučajna spremenljivka in je neodvisna od ostalih bitov. Naša domneva bo torej neparametrična, zato se bomo osredotočili na **neparametrične teste**.

Poljubno statistično hipotezo bomo preskusili z vzorcem $\boldsymbol{z} = (x_1, \dots, x_n)$, nato jo bomo zavrnili ali sprejeli ali pa se o njej ne bomo odločili. Ponavadi se odločimo za sklepanje:

1. hipotezo H_0 ali zavrnemo ali o njej ne odločimo;

2. hipotezo H_0 ali zavrnemo ali sprejmemo.

Preskus hipoteze s prvim načinom odločanja imenujemo **preskus značilnosti**, z drugim načinom odločanja pa **statistični test** ali **statistični preskus**. V primeru o teži štruc uporabimo statistični test, ko pa preskušamo naključnost PRBG z različnimi statistikami, pa delamo preskuse značilnosti.

Množico vseh vzorcev označimo z W, množico tistih, pri katerih hipotezo H_0 pri danem pravilu odločanja zavrnemo, pa z W_0 in jo imenujemo **kritično območje**. Če zavrnemo hipotezo, ki je pravilna, pravimo, da smo zagrešili **napako prve vrste**. Če z α_{F_0} označimo verjetnost[10] za napako prve vrste pri pogoju, da je porazdelitvena funkcija F_X enaka neki izbrani porazdelitveni funkciji F_0, potem je

$$\alpha_{F_0} = P(\boldsymbol{Z} \in W_0 | F_X = F_0)\,.$$

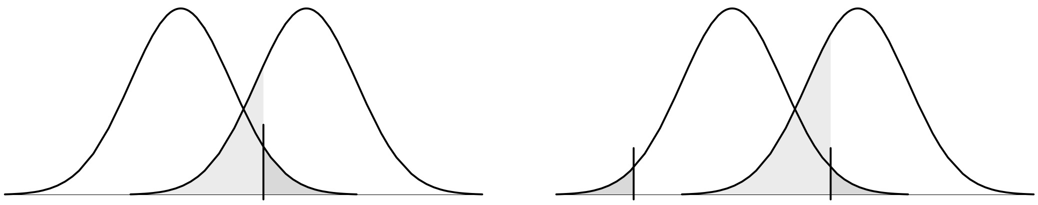
Z α sedaj označimo natančno zgornjo mejo za verjetnost, da bomo pri poljubnem pogoju, ki določa, da je hipoteza pravilna, torej $F_0 \in \mathcal{D}_0$, le to zavrnili. Ta zgornja meja se imenuje **stopnja značilnosti** preskušanja hipoteze in je enaka

$$\alpha = \sup_{F_0 \in \mathcal{D}_0} \alpha_{F_0}\,.$$

Če pa sprejmemo napačno hipotezo, oz. pri preskusu značilnosti o njej ne odločimo, pravimo, da smo zagrešili **napako druge vrste**. Ponavadi predpišemo stopnjo značilnosti $\alpha \in \mathbb{R}$ in potem določimo množico W_0 vzorcev, na podlagi katerih bomo hipotezo zavrnili. Kako velik α izberemo, je odvisno od posledic, ki bi jih imela zavrnitev pravilne hipoteze (napaka prve vrste). Seveda pa ne smemo izbrati premajhnega α, saj se s tem poveča možnost za napako druge vrste. V praksi ponavadi vzamemo α med 0,001 in 0,05.

[10] $P(A|B)$ je oznaka za pogojno verjetnost, to je verjetnost, da se je zgodil dogodek A, če vemo, da se je zgodil dogodek B, in jo izračunamo z enačbo $P(A|B) = P(A \cap B)\,/\,P(B)$. Včasih uporabimo oznako $P_B(A)$.

Primer 3.2.3. *Privzemimo, da vemo, da je slučajna spremenljivka X porazdeljena po zakonu $N(0,1)$ ali pa $N(2'5,1)$, kar pomeni $\mathcal{D} = \{N(0,1), N(2'5,1)\}$. Preskusimo ničelno hipotezo $H_0(F_X = N(0,1))$. S Sliko 10 je prikazano, kako lahko pri fiksni stopnji značilnosti neustrezna izbira intervala zaupanja poveča napako druge vrste. Ploščina temno osenčenega lika predstavlja napako prve vrste in s tem, ker je hipoteza enostavna, tudi stopnjo značilnosti, ploščina svetlo osenčenega lika pa napako druge vrste.*



Slika 10: Ustrezna in neustrezna izbira intervala zaupanja

Za izbrano stopnjo značilnosti α bi bila najboljša izbira kritičnega območja W_0 taka, da je pri vsakem pogoju, ki določa, da je hipoteza napačna ($F_X = F_1$, kjer je $F_1 \in \mathcal{D} - \mathcal{D}_0$), verjetnost za napako druge vrste

$$P(\boldsymbol{Z} \in W - W_0 | F_X = F_1)$$

najmanjša (glede na izbiro W_0). Test s tako izbranim kritičnim območjem se imenuje **enakomerno najmočnejši**, ni pa nujno, da sploh obstaja. Vprašanje, kdaj obstaja tak test, presega okvir te diplome, odgovor za nekaj posebnih primerov pa lahko najdemo v [8, 3. pogl.]. V nekaterih primerih obstajajo enakomerno najmočnejši testi pri pogoju, da W_0 izbiramo le iz množice tistih, ki določajo, da je test **nepristranski**. To pomeni, da je

$$P(\boldsymbol{Z} \in W_0 | F_X = F_0) \leq P(\boldsymbol{Z} \in W_0 | F_X = F_1)$$

za poljubno izbrana $F_0 \in \mathcal{D}_0$ in $F_1 \in \mathcal{D} - \mathcal{D}_0$. Z drugimi besedami, ni mogoče, da bi bila hipoteza z večjo verjetnostjo zavrnjena, pri nekem pogoju, ki pravi, da je pravilna, kot pri nekem drugem pogoju, ki pravi, da je napačna.

Primer 3.2.4. *Za normalno porazdeljeno slučajno spremenljivko X z $\operatorname{var}(X) = \sigma^2$ bomo preverjali domnevo, da je njeno matematično upanje enako nekemu realnemu številu a_0. Množica dopustnih porazdelitev bo torej*

$$\mathcal{D} = \{N(a, \sigma^2); a \in \mathbb{R}\},$$

domnevamo pa, da je $X \in \mathcal{D}_0$, kjer je $\mathcal{D}_0 = \{N(a_0, \sigma^2)\}$. Pravimo, da imamo pri normalni porazdelitvi hipotezo $H_0(a = a_0)$ proti hipotezi $H_1(a \neq a_0)$ pri znanem σ. Za preskus te hipoteze je v [8] pokazano, da enakomerno najmočnejši test ne obstaja, obstaja pa enakomerno najmočnejši nepristranski test velikosti α in sicer vzamemo kritično območje

$$W_0 = \left\{ \boldsymbol{z}; \frac{|\bar{x} - a_0|}{\sigma} \sqrt{n} \geq x_\alpha \right\},$$

kjer je x_α rešitev enačbe

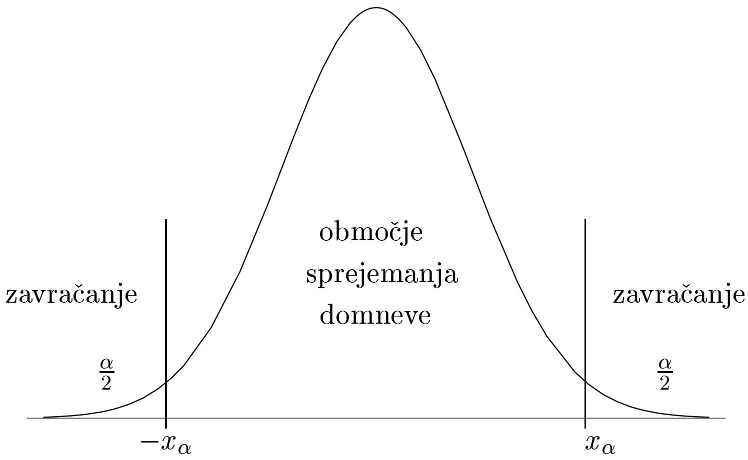
$$\frac{1}{\sqrt{2\pi}} \int_{-x_\alpha}^{x_\alpha} e^{-\frac{1}{2}u^2} du = 1 - \alpha. \tag{13}$$

Enačba pove, da je verjetnost, da standardno normalno porazdeljena slučajna spremenljivka zavzame vrednost na intervalu $[-x_\alpha, x_\alpha]$, enaka $1 - \alpha$.

Test v zgornjem primeru je opisan s testno funkcijo $U = \sqrt{n}(\bar{X} - a_0)/\sigma$. Vrednost x_α imenujemo **kritična vrednost** testne statistike. Ker hipotezo sprejmemo, oz. o njej ne odločimo takrat, ko je vrednost testne funkcije na intervalu $[-x_\alpha, x_\alpha]$, le tega poimenujemo **interval zaupanja**. V primeru, ko je hipoteza pravilna, torej ko je $X \sim N(a_0, \sigma^2)$, je $U \sim N(0,1)$. Tabela 2 predstavlja, kakšne intervale zaupanja vzamemo pri različnih stopnjah značilnosti testa. Dobimo jo iz enačbe (13), ki jo rešimo npr. s paketom za simbolno računanje *Mathematica*. Na Sliki 11 je pravilo odločanja tudi grafično prikazano.

α	0'2	0'1	0'05	0'02	0'01	0'005	0'002	0'001
x_α	1'2816	1'6449	1'9600	2'3263	2'5758	2'8070	3'0902	3'2905

Tabela 2: Kritične vrednosti x_α pri različnih stopnjah α značilnosti testa za $N(0,1)$ porazdelitev



Slika 11: Kritično območje

V nekaterih primerih je pri pogoju, da je domneva enostavna in resnična, testna statistika porazdeljena po zakonu $\chi^2(\nu)$. Ponavadi je taka domneva dobljena iz ocene variance. Tu nas majhno odstopanje od povprečja ne moti, zato vzamemo interval zaupanja, ki predstavlja kritično območje, oblike $[0, x_\alpha]$ in ne najkrajši možen pri dani stopnji značilnosti testa. V Tabeli 3 imamo pri $\chi^2(\nu)$ porazdelitvi za dane α določene x_α, kjer je x_α rešitev enačbe

$$\int_0^{x_\alpha} \frac{1}{\Gamma(\nu/2)2^{\nu/2}}\, x^{\nu/2-1} e^{-x/2} = 1 - \alpha\,. \tag{14}$$

V tabeli je za dan ν in α izračunan x_α, predstavljena pa je le za tiste parametre ν, ki se najpogosteje uporabljajo pri osnovnih testih PRBG.

Če je pri pogoju, da je domneva enostavna in resnična, testna statistika porazdeljena po zakonu $I(0,1)$, še ne vemo, kakšen interval zaupanja je potrebno vzeti, saj je le ta odvisen od **alternativne hipoteze** $H(F_X \in \mathcal{D} - \mathcal{D}_0)$. Včasih vzamemo interval zaupanja $[\frac{\alpha}{2}, 1 - \frac{\alpha}{2}]$, včasih pa $[0, 1 - \alpha]$.

3.3 NEPARAMETRIČNI TESTI

V tem razdelku sta opisana dva neparametrična testa za preskušanje neparametričnih hipotez. Pri opisih so navedene prednosti posameznega testa, na koncu razdelka pa je narejena tudi

$\nu\backslash\alpha$	0'1	0'05	0'025	0'01	0'005	0'0025	0'001
1	2'7055	3'8415	5'0239	6'6349	7'8794	9'1406	10'8276
2	4'6052	5'9915	7'3778	9'2103	10'5966	11'9829	13'8155
3	6'2514	7'8147	9'3484	11'3449	12'8382	14'3203	16'2662
7	12'0170	14'0671	16'0128	18'4753	20'2777	22'0404	24'3219
8	13'3616	15'5073	17'5345	20'0902	21'9550	23'7745	26'1245
9	14'6837	16'9190	19'0228	21'6660	23'5894	25'4625	27'8772
15	22'3071	24'9958	27'4884	30'5779	32'8013	34'9496	37'6973
31	41'4217	44'9853	48'2319	52'1914	55'0027	57'6923	61'0983

Tabela 3: Kritične vrednosti x_α pri različnih stopnjah α značilnosti testa za $\chi^2(\nu)$ porazdelitev

primerjava med njima. Za vsakega je napisano, kdaj ga lahko uporabimo, matematično ozadje ter postopek, kako ga uporabimo. Testa sta namenjena preskušanju hipoteze

$$H_0(F_X = F_0)\,,$$

da ima slučajna spremenljivka porazdelitveno funkcijo F_0. Takim testom pravimo **prilagoditveni testi** ali **testi soglasja**. χ^2-test bomo predstavili le za primer, ko je hipoteza H_0 enostavna, torej so vsi parametri porazdelitve F_0 dani, recimo $H_0(X \sim N(0,1))$.

3.3.1 χ^2-test

Razdelimo realno os na r razredov

$$E_1, \dots, E_r\,,$$

torej na r paroma disjunktnih podmnožic, in sicer tako da so verjetnosti

$$p_k = P_{H_0}(X \in S_k)$$

za vsak $k = 1, \dots, r$ pozitivne. Naj bo $(X_1, \dots, X_n)$ slučajni vzorec za slučajno spremenljivko X in

$$\boldsymbol{N} = (N_1, \dots, N_r)$$

frekvenčna porazdelitev vzorca po razredih $E_1, \dots, E_r$. Z N_k torej označimo število elementov vzorca v razredu E_k. Matematično upanje vektorja $(N_1, \dots, N_r)$ pri pogoju, da je hipoteza pravilna je

$$E_{H_0}(N_1, \dots, N_r) = (np_1, \dots, np_r)\,,$$

odstopanje eksperimentalnih frekvenc $(N_1, \dots, N_r)$ od hipotetičnih $(np_1, \dots, np_r)$ pa bi lahko ocenili z vsoto kvadratov

$$(N_1 - np_1)^2 + \dots + (N_r - np_r)^2\,.$$

Ta ocena ni dobra, saj smo realno os z nekim namenom razdelili na takšne razrede E_k, kot so, in sicer tako da je vsak enako pomemben. Mera odstopanja, ki to upošteva, je statistika

$$\chi^2 = \sum_{i=1}^{r} \frac{(N_i - np_i)^2}{np_i}\,,$$

ki jo imenujemo **Pearsonov 'hi kvadrat'**.

Izrek 3.3.1. *Pri pogoju, da je hipoteza pravilna, je limitna porazdelitev statistike χ^2 enaka $\chi^2(r-1)$.*

Dokaz. Definirajmo slučajne vektorje

$$\boldsymbol{X}^{(m)} = (X_1^{(m)}, \ldots, X_r^{(m)}), \qquad m = 1, \ldots, n,$$

tako da je vektor $\boldsymbol{X}^{(m)}$ **indikator izida** pri m-tem izbiranju, torej

$$X_k^{(m)} = \begin{cases} 1 & ; m\text{-ti element vzorca slučajne spremenljivke T je v množici } E_k \\ 0 & ; \text{sicer} \end{cases} .$$

Torej so vse komponente vektorja $\boldsymbol{X}^{(m)}$, razen ene, enake 0. Velja

$$\boldsymbol{N} = \boldsymbol{X}^{(1)} + \cdots + \boldsymbol{X}^{(n)}.$$

Vpeljimo še slučajne spremenljivke

$$Y_k = \frac{N_k - np_k}{\sqrt{np_k}}, \qquad k = 1, \ldots, r,$$

in iz njih naredimo slučajen vektor $\boldsymbol{Y} = (Y_1, \ldots, Y_r)$. Določimo sedaj karakteristično funkcijo vektorja $\boldsymbol{Y}$:

$$\phi_{\boldsymbol{Y}}(\boldsymbol{t}) = E\left(e^{i\sum_{k=1}^r t_k Y_k}\right) = e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} E\left(e^{i\sum_{k=1}^r t_k \frac{N_k}{\sqrt{np_k}}}\right).$$

Če upoštevamo $N_k = X_k^{(1)} + \cdots + X_k^{(n)}$, dobimo

$$\begin{aligned}
\phi_{\boldsymbol{Y}}(\boldsymbol{t}) &= e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} E\left(e^{i\sum_{k=1}^r \sum_{m=1}^n t_k \frac{X_k^{(m)}}{\sqrt{np_k}}}\right) = \\
&= e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} E\left(e^{i\sum_{m=1}^n \sum_{k=1}^r t_k \frac{X_k^{(m)}}{\sqrt{np_k}}}\right) = \\
&= e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} E\left(\prod_{m=1}^n e^{i\sum_{k=1}^r t_k \frac{X_k^{(m)}}{\sqrt{np_k}}}\right)
\end{aligned}$$

in ker je vzorec za slučajno spremenljivko T slučajen, je katerakoli slučajna spremenljivka $X_k^{(m)}$ neodvisna od poljubne kombinacije ostalih $X_i^{(j)}$, ki imajo zgornji indeks različen od m. Za neodvisni spremenljivki pa vemo, da je upanje njunega produkta enako produktu posameznih upanj, zato je

$$\phi_{\boldsymbol{Y}}(\boldsymbol{t}) = e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} \prod_{m=1}^n E\left(e^{i\sum_{k=1}^r t_k \frac{X_k^{(m)}}{\sqrt{np_k}}}\right).$$

Ker dogodki $\{X_k^{(m)} = 1; k = 1, \ldots, r\}$ predstavljajo kompleten sistem (pri poljubnem izidu se zgodi natanko en dogodek iz sistema) je

$$\phi_{\boldsymbol{Y}}(\boldsymbol{t}) = e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} \prod_{m=1}^n \sum_{k=1}^r P(X_k^{(m)} = 1) E\left(e^{i\sum_{k=1}^r t_k \frac{X_k^{(m)}}{\sqrt{np_k}}} | X_k^{(m)} = 1\right).$$

Za nek k je $X_k^{(m)} = 1$, ostali $X_i^{(m)}$, $i \neq k$, pa so enaki 0. To pomeni

$$\begin{aligned}\phi_{\boldsymbol{Y}}(\boldsymbol{t}) &= e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} \prod_{m=1}^n \sum_{k=1}^r p_k E\left(e^{it_k \frac{1}{\sqrt{np_k}}}\right) = \\ &= e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} \prod_{m=1}^n \sum_{k=1}^r p_k e^{it_k \frac{1}{\sqrt{np_k}}} = \\ &= e^{-i\sum_{k=1}^r t_k\sqrt{np_k}} \left(\sum_{k=1}^r p_k e^{it_k \frac{1}{\sqrt{np_k}}}\right)^n .\end{aligned}$$

Zadnjo enačbo logaritmiramo:

$$\ln \phi_{\boldsymbol{Y}}(\boldsymbol{t}) = -i\sum_{k=1}^r t_k\sqrt{np_k} + n\ln\sum_{k=1}^r p_k e^{it_k \frac{1}{\sqrt{np_k}}} .$$

Če člene $e^{it_k \frac{1}{\sqrt{np_k}}}$ razvijemo v Taylorjevo vrsto in pri zapisu uporabimo asimptotski simbol $\mathcal{O}$ iz analize, dobimo

$$\sum_{k=1}^r p_k e^{it_k \frac{1}{\sqrt{np_k}}} = 1 + \frac{i}{\sqrt{n}}\sum_{k=1}^r t_k\sqrt{p_k} - \frac{1}{2n}\sum_{k=1}^r t_k^2 + \mathcal{O}\left(\frac{1}{n\sqrt{n}}\right) .$$

Če je n dovolj velik, uporabimo še razvoj $\ln(1+h) = h - \frac{h^2}{2} + \mathcal{O}(h^3)$:

$$\ln\sum_{k=1}^r p_k e^{it_k \frac{1}{\sqrt{np_k}}} = \frac{i}{\sqrt{n}}\sum_{k=1}^r t_k\sqrt{p_k} - \frac{1}{2n}\sum_{k=1}^r t_k^2 + \frac{1}{2n}\left(\sum_{k=1}^r t_k\sqrt{p_k}\right)^2 + \mathcal{O}\left(\frac{1}{n\sqrt{n}}\right) .$$

Slednje vstavimo v zadnjo enačbo za $\phi_{\boldsymbol{Y}}$:

$$\ln \phi_{\boldsymbol{Y}}(\boldsymbol{t}) = -\frac{1}{2}\sum_{k=1}^r t_k^2 + \frac{1}{2}\left(\sum_{k=1}^r t_k\sqrt{p_k}\right)^2 + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) .$$

Torej je

$$\lim_{n\to\infty} \phi_{\boldsymbol{Y}}(\boldsymbol{t}) = e^{-\frac{1}{2}(t_1^2+\cdots+t_r^2)+\frac{1}{2}(t_1\sqrt{p_1}+\cdots+t_r\sqrt{p_r})^2} .$$

Uporabimo izrek (2.5.3), ki govori o konvergenci karakterističnih funkcij in pravi, da slučajni vektor $\boldsymbol{Y}$ konvergira v porazdelitvi k slučajni spremenljivki, ki ima zgornjo limito za karakteristično funkcijo. Pokažimo, da je

$$\phi_{\widetilde{\boldsymbol{Y}}} = e^{-\frac{1}{2}(t_1^2+\cdots+t_r^2)+\frac{1}{2}(t_1\sqrt{p_1}+\cdots+t_r\sqrt{p_r})^2} \tag{15}$$

karakteristična funkcija nekega večrazsežno normalno porazdeljenega slučajnega vektorja $\widetilde{\boldsymbol{Y}}$. Vzemimo poljubno $r \times r$ dimenzionalno ortogonalno matriko $\boldsymbol{A}$, katere spodnja vrstica je enaka $(\sqrt{p_1}, \ldots, \sqrt{p_r})$. Vpeljimo novo spremenljivko $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{t}$. Posledica ortogonalnosti je ohranjanje norme:

$$t_1^2 + \cdots + t_r^2 = u_1^2 + \cdots + u_r^2 .$$

Iz zadnje vrstice matrike $\boldsymbol{A}$ in predpisa $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{t}$ dobimo

$$u_r = \sqrt{p_1}t_1 + \cdots + \sqrt{p_r}t_r .$$

Zadnji dve relaciji vstavimo v enačbo (15):

$$\phi_{\widetilde{\boldsymbol{Y}}}(\boldsymbol{A}^T\boldsymbol{u}) = e^{-\frac{1}{2}(u_1^2+\cdots+u_{r-1}^2)} = \phi_{N_r(\boldsymbol{0},\boldsymbol{I}_{r-1})}(\boldsymbol{u})\,.$$

Poglejmo, kaj nam predstavlja $\phi_{\widetilde{\boldsymbol{Y}}}(\boldsymbol{A}^T\boldsymbol{u})$:

$$\phi_{\widetilde{\boldsymbol{Y}}}(\boldsymbol{A}^T\boldsymbol{u}) = E(e^{i(\boldsymbol{A}^T\boldsymbol{u})^T\widetilde{\boldsymbol{Y}}}) = E(e^{i\boldsymbol{u}^T\boldsymbol{A}\widetilde{\boldsymbol{Y}}}) = \phi_{\boldsymbol{A}\widetilde{\boldsymbol{Y}}}(\boldsymbol{u})\,.$$

To pomeni, da je $\boldsymbol{A}\widetilde{\boldsymbol{Y}} \sim N_r(\boldsymbol{0},\boldsymbol{I}_{r-1})$. Naredimo le še sklepno misel za dokaz izreka:

$$\overset{d}{\lim_{n\to\infty}} \chi^2 = \overset{d}{\lim_{n\to\infty}} \|\boldsymbol{Y}\|^2\,.$$

Ker je norma zvezna funkcija, velja

$$\overset{d}{\lim_{n\to\infty}} \|\boldsymbol{Y}\|^2 = \|\overset{d}{\lim_{n\to\infty}} \boldsymbol{Y}\|^2 = \|\widetilde{\boldsymbol{Y}}\|^2$$

in ker ortogonalne preslikave ohranjajo normo, je

$$\|\widetilde{\boldsymbol{Y}}\|^2 = \|\boldsymbol{A}\widetilde{\boldsymbol{Y}}\|^2 = (\boldsymbol{A}\widetilde{\boldsymbol{Y}})_1^2 + \cdots + (\boldsymbol{A}\widetilde{\boldsymbol{Y}})_{r-1}^2\,,$$

saj smo pokazali, da je $\boldsymbol{A}\widetilde{\boldsymbol{Y}} \sim N_r(\boldsymbol{0},\boldsymbol{I}_{r-1})$, kar pomeni, da je prvih $r-1$ komponent tega vektorja neodvisnih in standardno normalno porazdeljenih, zadnja pa je enaka 0. Iz relacij (7) in (11) iz prejšnjega poglavja pa vidimo, da je

$$(\boldsymbol{A}\widetilde{\boldsymbol{Y}})_1^2 + \cdots + (\boldsymbol{A}\widetilde{\boldsymbol{Y}})_{r-1}^2 \sim \chi^2(r-1)\,,$$

in dokaz je končan. ∎

Prevelika vrednost Pearsonove χ^2-statistike, torej preveliko odstopanje eksperimentalnih frekvenc od hipotetično pričakovanih, navadno ni posledica slučajnega izbiranja vzorca, ampak tega, da je hipoteza $H_0(F_X = F_0)$ napačna. Zato si izberemo interval zaupanja oblike $[0, x_\alpha]$.

Porazdelitev Pearsonove χ^2-statistike je odvisna od porazdelitve slučajne spremenljivke X, zato bi jo morali za vsak primer posebej določiti. To velja le za majhne vzorce, za velike pa lahko uporabimo limitno porazdelitev, kritično vrednost x_α pa takrat določimo iz Tabele 3, oz. enačbe (14), ki sta obe v prejšnjem razdelku. Kdaj je porazdelitev Pearsonove χ^2-statistike dovolj blizu limitni,[11] je odvisno od hitrosti konvergence, ta pa je odvisna od

1. števila n, to je velikosti vzorca,
2. števila r, to je števila razredov $E_1, \ldots, E_r$ ter od
3. parametrov $p_1, \ldots, p_r$, torej "velikosti razredov".

Zato ni točnega pravila, kdaj je n dovolj velik. Navadno je takrat, ko so vsi produkti $np_1, \ldots, np_r$ veliki vsaj 5. Ta zahteva je v splošnem prestroga. Če je razčlenitev na razrede enakomerna,

$$p_1 = \ldots = p_r = \frac{1}{r}\,,$$

[11] Reference o hitrosti konvergence te porazdelitve so navedene v [8, str. 254].

takrat polinomska porazdelitev vektorja $(N_1, \dots, N_r)$ najhitreje konvergira k normalni in lahko uporabimo limitno porazdelitev Pearsonove χ^2-statistike že za vzorce velikosti 15–20, če so le vse eksperimentalne frekvence $n_1, \dots, n_r$ velike vsaj 1, kjer je $(n_1, \dots, n_r)$ realizacija slučajnega vektorja $(N_1, \dots, N_r)$.

Pri preverjanju statistične hipoteze pravimo, da je test **dosleden**, če je za velike n verjetnost, da test zavrne hipotezo H_0, ko je ta nepravilna in H_1 pravilna, poljubno približa k 1. Kljub temu, da χ^2-test ni enakomerno najmočnejši, je dosleden proti vsaki alternativni hipotezi H_1, ki da drugačno porazdelitev verjetnosti $(p_1, \dots, p_r)$ kot hipoteza H_0. Za dosledne teste velja tudi, da so asimptotično nepristranski.

Če je porazdelitev F_0 zvezna in razčlenitev realne osi enakomerna, se za vzorec velikosti n priporoča vzeti število razredov r_n, tako da je

$$r_n \sim 2\sqrt[5]{2\frac{n^2}{d_\alpha^2}}\,,$$

pri čemer d_α ustreza enačbi

$$\frac{1}{\sqrt{2\pi}}\int_{d_\alpha}^{\infty} e^{-\frac{1}{2}u^2}du = \alpha$$

in ga lahko izračunamo s pomočjo Tabele 2, glej [8, str. 254]. Pri $n = 100$ in $\alpha = 0,05$ torej lahko vzamemo $r_n = 12$.

Ponavadi iz vrednosti Pearsonove χ^2-statistike določimo vrednost $p \in [0, 1)$, ki nam pove, s kolikšno verjetnostjo bi pri predpostavki, da je hipoteza $H_0(F_X = F_0)$ pravilna, dobili slučajni vzorec, ki od pričakovane limitne frekvenčne porazdelitve odstopa manj,

$$p = \int_0^{\chi^2} f_{\chi^2(r-1)}(t)dt\,.$$

Pri predpostavki, da je hipoteza $H_0(F_X = F_0)$ pravilna, dobimo iz trditve (2.3.3), da je slučajna spremenljivka p enakomerno porazdeljena na intervalu $(0, 1)$. Če za preskušanje naše hipoteze test χ^2-ponovimo na več različnih med seboj neodvisnih slučajnih vzorcih, potem nam tako dobljene vrednosti p zopet predstavljajo slučajen vzorec za po zakonu $I(0, 1)$ porazdeljeno slučajno spremenljivko in lahko preskusimo še to hipotezo. Za preskus hipoteze $H_0(p \sim I(0, 1))$ ponavadi uporabimo test Kolmogorova.

Postopek izvedbe χ^2-testa

```
▷ določi slučajni vzorec (X_1, ..., X_n) slučajne spremenljivke X : D → ℝ
▷ določi particijo ℝ = E_1 ∪ E_2 ∪ ... ∪ E_r
▷ za k = 1 do r
    p_k = P_{H_0}(X ∈ E_k)
    N_k = število elementov slučajnega vzorca, ki so v E_k
▷ določi vrednost testne funkcije χ^2
  in njej ustrezno ``vrednost p''
    χ^2 = Σ_{k=1}^r (N_k − np_k)^2 / (np_k)
    p = 1/(Γ((r − 1)/2) 2^{(r−1)/2}) ∫_0^{χ^2} t^{(r−1)/2−1} e^{−t/2}(t) dt
▷ rezultat:  Pri predpostavki, da je hipoteza H_0(F_X = F_0) pravilna,
    je verjetnost, da dobimo slučajni vzorec, ki še bolj odstopa od
    pričakovane limitne frekvenčne porazdelitve, enaka 1 − p.
    Če je 1 − p < α, hipotezo zavrnemo.
```

Primer 3.3.2. *Radi bi ugotovili, ali je igralna kocka poštena. Zato na rezultatih* 1000 *metov naredimo* χ^2*-test. Enko smo vrgli* 170*-krat, dvojko* 166*-krat, trojko* 185*-krat, štirico* 162*-krat, petko* 151*-krat in šestico* 166*-krat. Spremenljivka* X *nam predstavlja število pik po metu kocke. Za particijo realne osi bomo vzeli* $E_1 = (-\infty, 1]\,,\quad E_2 = (1,2]\,,\quad E_3 = (2,3]\,,\quad E_4 = (3,4]\,,\quad E_5 = (4,5]$ *in* $E_6 = (5,\infty)\,,$ *torej je frekvenčna porazdelitev po razredih enaka* $n_1 = 170\,,\quad n_2 = 166\,,\quad n_3 = 185\,,\quad n_4 = 162\,,\quad n_5 = 151$ *in* $n_6 = 166\,.$ *Ker preskušamo hipotezo*

$$H_0\left(X \sim \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}\right),$$

je $p_1 = \ldots = p_6 = \frac{1}{6}$*, kjer je* $p_i = P(X \in E_i)$*. Vrednost statistike* χ^2 *je potem* $3'692$*. Pri stopnji zaupanja* $\alpha = 0,05$ *je za* $\chi^2(5)$ *porazdeljeno slučajno spremenljivko komplement kritičnega območja enak* $[0, 11'0705)$*, torej našo hipotezo sprejmemo.*

3.3.2 Test Kolmogorova in Smirnova

V Diehard testih za PRBG srečamo oznako KS test, ki izhaja iz priimkov avtorjev dveh zelo podobnih testov. Ideja in matematične osnove so skoraj identične pri obeh testih, le uporaba je različna. Test Kolmogorova preskuša enostavno hipotezo $H_0(F_X = F_0)$, kjer je porazdelitvena funkcija F_0 točno določena in je zvezna. Test Smirnova pa preskuša hipotezo $H_0(F_X = F_Y)$, kjer sta X in Y slučajni spremenljivki, o katerih vemo le to, da imata zvezno porazdelitev, ne vemo pa kakšno.

Naj bo X slučajna spremenljivka in $(X_1, \ldots, X_n)$ njen slučajni vzorec. Z V_n označimo empirično porazdelitveno funkcijo vzorca,

$$V_n(x) = \frac{\text{število elementov v vzorcu, ki so manjši ali enaki } x}{n},$$

z D_n pa "sup-normo" razlike med porazdelitveno funkcijo F_0 in porazdelitveno funkcijo slučajnega vzorca za X, to je

$$D_n = \sup_{-\infty < x < \infty} |V_n(x) - F_0(x)|\,.$$

Kaj nam pove $|F_X(x) - F_Y(x)|$? Označimo ta izraz z d. Očitno je $d \in [0,1]$, pove pa, da je verjetnost, da je slučajna spremenljivka Y manjša od x, za d manjša ali večja od verjetnosti, da je slučajna spremenljivka X manjša od x. Recimo, da je $d = 0,3$ za $x = 5$. To pomeni, da je verjetnost $P(X \le x)$ za 30% recimo manjša od verjetnosti $P(Y \le x)$. Sklepali bi, da X in Y najbrž nista enako porazdeljeni. Toda podatek d je lahko zavajajoč. To se zgodi, kadar so vrednosti spremenljivk grupirane po nekih majhnih intervalih ali pa celo diskretne. Recimo, da velja $P(X = 1000) = 1$ in $P(Y = 1001) = 1$. Porazdelitvi spremenljivk X in Y sta skoraj enaki, toda $D = \sup_x d = 1$ je maksimalen možen. Če zahtevamo, da sta porazdelitvi F_X in F_Y zvezni, se nam to skoraj ne more zgoditi. To je razlog, da je test Kolmogorova uporaben le, če je X zvezna slučajna spremenljivka. Lahko pa bi bile vrednosti spremenljivk grupirane po nekih majhnih intervalih in imamo zopet isti problem. Odpravimo ga s tem, da za razliko med porazdelitvenima funkcijama namesto sup-norme uporabimo integralsko normo. Ta ne upošteva "kratkotrajnih skokov" razlike med funkcijama. To je storjeno v Anderson-Darlingovi izboljšavi testa Kolmogorova.

Iz Glivenkovega izreka vemo, da je

$$P_{H_0}(\overset{d}{\lim_{n\to\infty}} D_n = 0) = 1\,,$$

toda nas zanima način (asimptotičnost) te konvergence. Odgovor dobimo iz Kolmogorovega izreka, ki nam podaja limitno porazdelitev $Q = \lim^d_{n\to\infty} Q_n$ porazdelitvene funkcije slučajne spremenljivke $Q_n = D_n\sqrt{n}$,

$$Q_n(\lambda) = \begin{cases} P(D_n\sqrt{n} < \lambda) & ; \lambda > 0 \\ 0 & ; \text{sicer} \end{cases} .$$

Izrek 3.3.3. (**Izrek Kolmogorova.**) *Če je porazdelitvena funkcija F_0 zvezna in hipoteza $H_0(F_X = F_0)$ pravilna, je za vsak pozitiven λ limitna porazdelitvena funkcija spremenljivke $D_n\sqrt{n}$ enaka*

$$Q(\lambda) = \begin{cases} \sum_{k=-\infty}^{\infty}(-1)^k e^{-2k^2\lambda^2} & ; \lambda > 0 \\ 0 & ; \text{sicer} \end{cases} . \qquad \blacksquare$$

Dokaz izreka je zelo zahteven, zato ga bomo izpustili, navedli pa bomo nekaj referenc, kje ga najdemo. Izrek je prvi dokazal Kolmogorov[12], krajši dokaz je našel Feller[13], najelegantnejšega pa Doob[14]. Skico Doobovega dokaza najdemo v [5, str. 396], kjer vidimo, da je ideja dokaza v tem, da problem prevedemo na stohastične procese.

Ko bomo uporabljali test Kolmogorova na vzorcih iz PRBG, bodo najbolj zaskrbljujoče ekstremne vrednosti, zato jim bomo dali večjo težo, kar pomeni, da jih bomo pomnožili z nekim velikim pozitivnim številom. Tako bo test na vzorcu, ki naj bi bil enakomerno porazdeljen na intervalu $[0, 1]$, bolj zaznal grupiranje na robovih (npr. vzorec (0'1,0'2,0'9)) kot pa grupiranje na sredini intervala (npr. vzorec (0'3,0'5,0'5)). Takšno različico testa Kolmogorova sta skonstruirala Anderson in Darling in je opisana v članku [1], kjer je tudi izdelan dokaz s pomočjo Donskerjevega[15] izreka. To je izrek, ki ga potrebujemo pri Doobovem dokazu izreka Kolmogorova. Mi bomo v naslednji trditvi povzeli le za nas uporaben del teorije iz članka. To je tisti del, ko vzamemo integralsko normo in utež, kot jo uporablja KS test v Diehard testih.

Trditev 3.3.4. *Naj bo X slučajna spremenljivka in $(X_1, \ldots, X_n)$ njen slučajni vzorec. Z V_n označimo empirično porazdelitveno funkcijo vzorca,*

$$V_n(x) = \frac{\textit{število elementov v vzorcu, ki so manjši ali enaki } x}{n} ,$$

z D_n pa uteženo integralsko 2-normo razlike med porazdelitveno funkcijo F_0 in porazdelitveno funkcijo slučajnega vzorca za X,

$$D_n^2 = \int_{-\infty}^{\infty} [V_n(x) - F_0(x)]^2 \psi[F_0(x)] dF_0(x) ,$$

kjer je utež $\psi : [0, 1] \to \mathbb{R}$ neka p.p. zvezna (v vseh, razen v končno mnogo točkah ne) pozitivna funkcija. Brez škode za splošnost lahko privzamemo, da je F_0 porazdelitvena funkcija $I(0, 1)$ porazdelitve. Če ni, potem uporabimo trditev (2.3.3) in gremo preverjati hipotezo $H_0(F_0(X) \sim I(0, 1))$. Torej je $F_0(x) = x \cdot \chi_{[0,1]}(x)$ in zgornja definicija D_n se poenostavi v

$$D_n^2 = \int_0^1 [V_n(x) - x]^2 \psi(x) dx .$$

[12] A. N. Kolmogorov, *Sulla determinazione empirica di una legge di distribuzione*, Giorn. Ist. Ital. Attuari, **4** (1933), str. 83.

[13] W. Feller, *On the Kolmogorov-Smirnov limit theorems for empirical distributions*, Annals of Mathematical Statistics, **19** (1948), str. 177.

[14] J. L. Doob, *Heuristic approach to the Kolmogorov-Smirnov theorems*, Annals of Mathematical Statistics, **20** (1949), str. 393.

[15] M. D. Donsker, *Justification and extension of Doob's heuristic approach to the Kolmogorov-Smirnov theorems*, Annals of Mathematical Statistics, **23** (1952), 277–281.

S Q_n označimo porazdelitveno funkcijo slučajne spremenljivke nD_n^2,

$$Q_n(\lambda) = P(nD_n^2 \le \lambda)\,.$$

Po Donskerjevem izreku obstaja limita $\lim^d_{n\to\infty} Q_n(\lambda)$, zato za vsak λ velja enačba

$$\lim_{n\to\infty} P\left\{\int_0^1 n[V_n(x)-x]^2\psi(x)dx \le \lambda\right\} = P\left\{\int_0^1 \left(\lim_{n\to\infty} n[V_n(x)-x]^2\right)\psi(x)dx \le \lambda\right\},$$

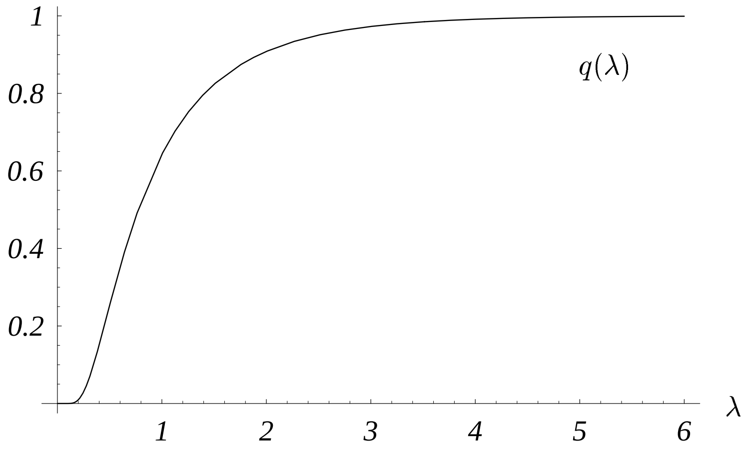
kjer je limita na desni strani porazdelitev zvezne slučajne spremenljivke. Desno stran enačbe lahko pri dani uteži izračunamo. Najzanimivejša sta primera, ko ni uteži ($\psi(x) = 1$) ali ko obtežimo interval $(0,1)$ z utežjo

$$\psi(x) = \frac{1}{x(1-x)}\,,$$

ki ima največje vrednosti ravno na robovih intervala $(0,1)$. V drugem primeru se prejšnja enačba pretvori v

$$\lim_{n\to\infty} P(nD_n^2 \le \lambda) = \frac{\sqrt{2\pi}}{\lambda}\sum_{j=0}^{\infty}\binom{-\frac{1}{2}}{j}(4j+1)e^{-(4j+1)^2\pi^2/(8\lambda)} \times$$
$$\times \int_0^{\infty} e^{\lambda/(8(w^2+1))-(4j+1)^2\pi^2w^2/(8\lambda)}dw\,.$$

Kakšna je limitna funkcija $q(\lambda) = \lim_{n\to\infty} P(nD_n^2 \le \lambda)$, si lahko ogledamo na Sliki 12, za numerične izračune pa jo aproksimiramo z neko odsekoma polinomsko funkcijo.



Slika 12: graf funkcije $q(\lambda)$ za Anderson-Darling različico testa Kolmogorova

Za običajen test Kolmogorova mora biti vzorec toliko velik, da se točna porazdelitev statistike $D_n\sqrt{n}$ dovolj približa njeni limitni porazdelitvi. Pri običajnih stopnjah značilnosti med 0,01 in 0,05 je to tedaj, ko je v vzorcu vsaj 80 elementov.

Test Kolmogorova je zaradi Glivenkovega izreka dosleden in je močnejši od testa χ^2, saj ne izgubi nobene informacije o vzorcu, ker ga ne grupira. Po drugi strani pa je test Kolmogorova časovno zahtevnejši. Proti testu χ^2 ima še to pomankljivost, da mora biti porazdelitvena funkcija F_0 zvezna.

Postopek izvedbe Anderson-Darling različice testa Kolmogorova z utežjo $\psi(x) = \frac{1}{x(1-x)}$
▷ `določi slučajni vzorec` $(X_1, \ldots, X_n)$ `slučajne spremenljivke` X `za preskus hipoteze` $H_0(X \sim I(0,1))$ ▷ `določi empirično porazdelitev vzorca` $V_n(x) = \dfrac{\text{število elementov v vzorcu, ki so manjši ali enaki } x}{n}\,,$ ▷ `določi kvadrat utežene 2-integralske norme razlike med` `dejansko in empirično porazdelitveno funkcijo` $D_n^2 = \sum_{i=0}^{n} \int_{X_i}^{X_{i+1}} \dfrac{(\frac{i}{n} - x)^2}{x(1-x)} dx\,,$ `kjer je` $X_0 = 0$ `in` $X_{n+1} = 1$ ▷ `definirajmo funkcijo` $q(\lambda) = \lim_{n\to\infty} P\{nD_n^2 \le \lambda\}$ `in iz` $q(\lambda) = \dfrac{\sqrt{2\pi}}{\lambda} \sum_{j=0}^{\infty} \binom{-\frac{1}{2}}{j} (4j+1) e^{-(4j+1)^2\pi^2/(8\lambda)} \int_0^{\infty} e^{\lambda/(8(w^2+1)) - (4j+1)^2\pi^2 w^2/(8\lambda)} dw$ `določi ``vrednost` $p$`''` $p = q(nD_n^2)$ ▷ `rezultat: Pri predpostavki, da je hipoteza` $H_0(X \sim I(0,1))$ `pravilna,` `je verjetnost, da dobimo slučajni vzorec, pri katerem empirična` `porazdelitvena funkcija` $V_n$ `še bolj odstopa od pričakovane` $F_0(x) = x$ `porazdelitvene funkcije, enaka` $1 - p$. `Če je` $1 - p < \alpha$`, hipotezo zavrnemo.`

4 STATISTIČNI TESTI ZA (P)RBG

To poglavje opisuje testne funkcije današnjih programov za testiranje naključnosti generatorjev (psevdo)naključnih zaporedij bitov. V prvem razdelku so opisani osnovni testi – vsakdo, ki bi poskusil raziskati lastnosti nekega zaporedja bitov, bi najprej pogledal, če ima zaporedje približno toliko enk kot ničel, če se pari enk in ničel pojavijo približno enakokrat, če to drži tudi za trojke, četverke, ... , če je porazdelitev blokov (enke, ki se v zaporedju držijo skupaj) takšna, kot mora biti, če se zaporedje od nikjer naprej ne nadaljuje niti približno tako, kot se je začelo. Z osnovnimi testi preverimo, da naše zaporedje nima takšnih napak. Če pa se našega generatorja (psevdo)naključnih zaporedij bitov – (P)RBG – loti pravi kriptograf, bo proučil, kako deluje, in ga napadel na osnovi najdenih šibkih točk. Npr., ker algoritmi uporabljajo linearne preslikave in linearne pomične registre (LFSR), bo kriptograf preučil linearno kompleksnost našega algoritma, o kateri smo pisali v 1. pogl., test linearnosti pa je 8. test v Crypt-XS testih. Seveda je prav, da preverjamo tudi neobičajne lastnosti, kot to delajo nekateri Diehard testi. Lahko bi rekli, da preverjajo poleg osnovnih tudi "naključne lastnosti", saj (P)RBG ponavadi nimajo osnovnih napak. V članku [13] avtor George Marsaglia pravi, da lahko vsak, ki ima nekaj znanja verjetnosti in statistike, dopolni zbirko že znanih testov s svojimi.

Primer 4.0.5. *Tako bi npr. lahko dano zaporedje naključnih bitov razdelili na* 15 *enako velikih kosov ter vsak kos razdelili na* 17 *bitov dolge koščke, ki bi nam predstavljali cela števila. Za vsak kos bi naredili zaporedje ostankov pri deljenju s* 23 *in na tem zaporedju* χ^2*-test in na danih* 15 *p-jih še KS test.*

Če (P)RBG testiramo s Crypt-XS ter Diehard testi, smo s tem pokrili vse osnovne teste. Zahteve standarda FIPS 140-1 pa so namenjene bolj inicializaciji (P)RBG in zato s šibkejšimi zahtevami preverjajo osnovne lastnosti kratkih sekvenc.

Za preverjanje kriptografske varnosti PRBG bi morali poznati že odkrite napade na PRBG in preveriti, če slučajno nima naš PRBG kakšne take šibkosti, ki jo izkorišča kateri od napadov. Nekaj teh napadov je opisanih v [15], za mnoge pa so v tem priročniku dane le reference, glej [15, str. 216–222]. Tako so za vsak primer nelinearizacije LFSR, ki so prikazane na Slikah 4, 5 in 6 na strani 19, dani v [15, str. 202–212] konkretni primeri, ki ohranijo dobre osnovne statistične lastnosti in zagotovijo nelinearnost izhodnih zaporedij, vendar pa jim ne uspe zagotoviti npr. neavtokoreliranosti.

Opis osnovnih testov je vzet iz [15] in [19], opis Diehard testov iz [13], [14] ter samem izpisu rezultatov testa, opis Crypt-XS testov pa iz navodil za uporabo [19], ki so priloženi programskemu paketu.

4.1 PET OSNOVNIH STATISTIČNIH TESTOV ZA (P)RBG

Razdelek opisuje pet osnovnih testov, navedenih v [15], ki se uporabljajo za testiranje naključnosti nekega zaporedja bitov $s = s_1, \dots, s_n$. Vsak od testov opisuje različne statistike, ki vsaka zase preskuša hipotezo, da je bilo zaporedje s generirano z RBG.

1. **Monobit test (frequency test)**
   Cilj tega testa je ugotoviti, ali ima dano zaporedje s približno toliko bitov enakih nič (ničel) kot tistih enakih ena (enk). Če z n_0 označimo število ničel, z n_1 pa število enk v zaporedju s, je statistika
   $$X_1 = \frac{(n_0 - n_1)^2}{n}$$

pri zaporedju, generiranim s pravim RBG, za dovolj velike n porazdeljena po zakonu $\chi^2(1)$.

Dokaz. Ker naj bi bili posamezni biti neodvisni med seboj, lahko naredimo χ^2-test, glej razdelek (3.3.1), za slučajno spremenljivko X, ki nam predstavlja vrednost bita. Na slučajnem vzorcu $(s_1, \dots, s_n)$ bomo preskusili hipotezo $X \sim \text{Bin}(1, \frac{1}{2})$. Pri oznakah, ki smo jih uporabili v razdelku (3.3.1), vzamemo $E_1 = (-\infty, 0]$ in $E_2 = (0, \infty)$. Potem je $p_1 = p_2 = \frac{1}{2}$ in Pearsonova χ^2-statistika

$$X_1 = \frac{(n_0 - n\frac{1}{2})^2}{n\frac{1}{2}} + \frac{(n_1 - n\frac{1}{2})^2}{n\frac{1}{2}} = \dots = \frac{(n_0 - n_1)^2}{n}$$

je potem asimptotično porazdeljena po zakonu $\chi^2(1)$. ∎

2. **Test parov** (**two-bit test** ali **serial test**)

   Cilj tega testa je ugotoviti, ali ima dano zaporedje približno enako število pojavitev podzaporedij `00`, `01`, `10` in `11`. Naj n_0 in n_1 zaporedoma označujeta število ničel in število enk ter n_{00}, n_{01}, n_{10} in n_{11} zaporedoma število pojavitev para 00, 01, 10 in 11. Očitno je $n_{00} + n_{01} + n_{10} + n_{11} = n - 1$, saj dovolimo prekrivanje podzaporedij. Porazdelitev statistike

$$X_2 = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n}(n_0^2 + n_1^2) + 1$$

   se pri zaporedju, generiranim s pravim RBG, za dovolj velike n ($n \geq 21$) približuje $\chi^2(2)$ porazdelitvi.

   Dokaz. Navedli bomo le skico dokaza, saj je sam dokaz predolg. V [3] pa lahko najdemo tudi izrek v bolj splošni obliki.

   Za $i = 1, 2, \dots, n-1$ označimo

$$\boldsymbol{\xi}_i = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \text{ če je } s_i = 0 \text{ in } s_{i+1} = 0 \quad , \qquad \boldsymbol{\xi}_i = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \text{ če je } s_i = 0 \text{ in } s_{i+1} = 1$$

$$\boldsymbol{\xi}_i = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \text{ če je } s_i = 1 \text{ in } s_{i+1} = 0 \quad \text{ter} \quad \boldsymbol{\xi}_i = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \text{ če je } s_i = 1 \text{ in } s_{i+1} = 1\,.$$

   Ni se težko prepričati, da so slučajni vektorji $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_{n-1}$ enako porazdeljeni in da velja

$$\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 + \ldots + \boldsymbol{\xi}_{n-1} = \begin{pmatrix} n_{00} \\ n_{01} \\ n_{10} \\ n_{11} \end{pmatrix} \quad , \qquad E(\boldsymbol{\xi}_1) = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix},$$

$$\text{var}(\boldsymbol{\xi}_1) = \begin{bmatrix} 3/16 & -1/16 & -1/16 & -1/16 \\ -1/16 & 3/16 & -1/16 & -1/16 \\ -1/16 & -1/16 & 3/16 & -1/16 \\ -1/16 & -1/16 & -1/16 & 3/16 \end{bmatrix} \quad \text{ter}$$

$$\text{cov}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) = \begin{bmatrix} -1/16 & 0 & 0 & 0 \\ 0 & -1/16 & 0 & 0 \\ 0 & 0 & -1/16 & 0 \\ 0 & 0 & 0 & -1/16 \end{bmatrix}.$$

Če označimo $\boldsymbol{\Sigma} = \operatorname{var}(\boldsymbol{\xi}_1)$ in $\boldsymbol{C} = \operatorname{cov}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ ter če upoštevamo, da sta slučajna vektorja $\boldsymbol{\xi}_i$ in $\boldsymbol{\xi}_j$ neodvisna za indeksa i in j, ki se po absolutni vrednosti razlikujeta za več kot 1, potem iz splošne formule za kovariančno matriko vsote slučajnih vektorjev,

$$\operatorname{var}\left(\sum_{i=1}^{n-1} \boldsymbol{\xi}_i\right) = \sum_{i=1}^{n-1} \operatorname{var}(\boldsymbol{\xi}_i) + 2\sum_{i<j} \operatorname{cov}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)\,,$$

dobimo

$$\operatorname{var}\left(\sum_{i=1}^{n-1} \boldsymbol{\xi}_i\right) = (n-1)\boldsymbol{\Sigma} + 2(n-2)\boldsymbol{C}\,.$$

Iz centralnega limitnega izreka za vektorje potem dobimo

$$\frac{\sum_{i=1}^{n-1} \boldsymbol{\xi}_i - \begin{pmatrix} \frac{n-1}{4} \\ \frac{n-1}{4} \\ \frac{n-1}{4} \\ \frac{n-1}{4} \end{pmatrix}}{\sqrt{n}} \xrightarrow{d} N(\mathbf{0}, \boldsymbol{\Sigma} + 2\boldsymbol{C})\,.$$

Nato pokažemo, da je $\boldsymbol{\Sigma} + 2\boldsymbol{C}$ idempotentna matrika z rangom 2, in ker velja

$$\sum_{i=1}^{n-1} \boldsymbol{\xi}_i \xrightarrow{d} \boldsymbol{S}$$

za nek slučajni vektor $\boldsymbol{S}$, ter zato tudi

$$\left\|\sum_{i=1}^{n-1} \boldsymbol{\xi}_i\right\|^2 \xrightarrow{d} \|\boldsymbol{S}\|^2\,,$$

je

$$\boldsymbol{S} \sim \chi^2(2)\,.$$

To pa limitna porazdelitev statistike X_2. ■

3. **Poker test**
Zaporedje s razdelimo na k delov dolžine m, kjer naj bo $k \geq 5 \cdot 2^m$. Tako dobimo zaporedje $z = z_1, \ldots, z_k$ in vsak njegov člen nam predstavlja neko število i v binarnem zapisu, $0 \leq i \leq 2^m - 1$. Z n_i označimo število pojavitev števila i v zaporedju z (npr. n_{14} je število pojavitev elementa `01110` v zaporedju z). Poker test preveri, ali se vsako število i pojavi v z približno enakokrat. Porazdelitev statistike

$$X_3 = \frac{2^m}{k}\left(\sum_{i=0}^{2^m-1} n_i^2\right) - k \tag{16}$$

je pri zaporedju, generiranim s pravim RBG, približno enaka $\chi^2(2^m - 1)$ porazdelitvi.

Dokaz. Brez škode lahko predpostavimo, da je število n deljivo z m. Ker predpostavljamo, da je vzorec m-teric slučajen, lahko na njem naredimo χ^2-test:

$$X_3 = \sum_{i=0}^{2^m-1} \frac{(n_i - k/2^m)^2}{k/2^m}\,.$$

Če števec kvadriramo, vsoto razdelimo na tri dele in upoštevamo $\sum_{i=0}^{2^m-1} n_i = k$, dobimo iskano statistiko. ■

Za $m = 1$ je pri enaki stopnji značilnosti α poker test ekvivalenten monobit testu.

4. **Bločni test (runs test)**
   Zaporedje s razdelimo na **bloke** in **vrzeli**, to je na maksimalno dolga podzaporedja enk oz. ničel (`1110001100010001010011101111100` bi tako razdelili na `111-000-11-000-1-000-1--0-1-00-111-0-1111-00`). Pričakovano število pojavitev bloka dolžine i v zaporedju s dožine n, generiranim z RBG, je $e_i = (n-i+3)/2^{i+2}$. Naj bo k enak največjemu i, za katerega je $e_i \geq 5$ (ker je e_i odvisen od n, je tudi k odvisen od n). Z B_i in V_i označimo število blokov in vrzeli v s dolžine i. Slučajna spremenljivka

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(V_i - e_i)^2}{e_i}$$

   se pri zaporedju, generiranim z RBG, za velike n približuje $\chi^2(2k-2)$ porazdelitvi.

   Dokaz spustimo, saj je še težavnejši od dokaza porazdelitve pri testu parov.

5. **Avtokorelacijski test (autocorrelation test)**
   Namen testa je preveriti podobnost med zaporedjem s in zaporedjem dobljenim s premikom zaporedja s za d mest (izbrisom prvih d členov), $1 \leq d \leq \lfloor n/2 \rfloor$. Z $A(d)$ označimo število bitov zaporedja s, ki niso enaki istoležnim bitom v premiku s, dobimo

$$A(d) = \sum_{i=1}^{n-d} s_i \oplus s_{i+d}\,,$$

   kjer $\oplus$ označuje XOR (ekskluzivni ali) operator. Tu se statistika

$$X_5 = 2\left(A(d) - \frac{n-d}{2}\right)/\sqrt{n-d}$$

   za velike n ($n - d \geq 10$) približuje $N(0,1)$ porazdelitvi.

   Dokaz. Če vzamemo

$$z_i = s_i \oplus s_{i+d}\,, \qquad i = 1, \ldots, n-d\,,$$

   lahko pokažemo, da so slučajne spremenljivke $z_1, \ldots, z_{n-d}$ med seboj neodvisne. Pokazati moramo

$$P(z_1 \in A_1, \ldots, z_{n-d} \in A_{n-d}) = P(z_1 \in A_1) \cdot \ldots \cdot P(z_{n-d} \in A_{n-d})\,,$$

   kjer za poljuben A_i lahko vzamemo katerokoli izmed množic $\{\}$, $\{0\}$, $\{1\}$ in $\{0,1\}$. Če smo pri kateremkoli i vzeli $A_i = \{\}$, sta obe strani enačbe enaki 0. Če pa smo vzeli $A_i = \{0,1\}$, se enačba trivialno poenostavi. Torej moramo pokazati

$$P(z_{n_1} = \alpha_{n_1}, \ldots, z_{n_k} = \alpha_{n_k}) = P(z_{n_1} = \alpha_{n_1}) \cdot \ldots \cdot P(z_{n_k} = \alpha_{n_k})\,,$$

   kjer je k naravno število, $k \leq n-d$, α_i enak 0 ali 1 in $\{n_1, \ldots, n_k\} \subset \{1, \ldots, n-d\}$. Desna stran enačbe je enaka $(1/2)^k$. Z indukcijo na k (število elementov zaporedja $n_1, \ldots, n_k$) pokažimo, da je tolikšna tudi leva stran. Izračunajmo najprej:

$$\begin{aligned}
&P(z_{n_1} = \alpha_{n_1}, \ldots, z_{n_k} = \alpha_{n_k}) = \\
&= P(s_{n_1} \oplus s_{n_1+d} = \alpha_{n_1}, \ldots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k}) = \\
&= \frac{1}{2} P(s_{n_1} \oplus s_{n_1+d} = \alpha_{n_1}, \ldots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k} | s_{n_1} = 0) + \\
&\quad + \frac{1}{2} P(s_{n_1} \oplus s_{n_1+d} = \alpha_{n_1}, \ldots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k} | s_{n_1} = 1) = \\
&= \frac{1}{2} P(s_{n_1+d} = \alpha_{n_1}, \ldots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k}) + \\
&\quad + \frac{1}{2} P(s_{n_1+d} = \bar{\alpha}_{n_1}, \ldots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k})\,,
\end{aligned}$$

kjer smo označili $\bar{\alpha_{n_1}} = 1 - \alpha_{n_1}$.

Označimo sklep, ki sledi z (*):
Spremenljivka s_{n_1+d} se v $P(s_{n_1+d} = \alpha_{n_1}, \dots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k})$ pojavi lahko le še v kombinaciji $s_{n_1+d} \oplus s_{n_1+2d} = \alpha_{n_1+d}$. Če se ne, dobimo iz neodvisnosti spremenljivk $s_1, \dots, s_n$

$$\begin{aligned} &P(z_{n_1} = \alpha_{n_1}, \dots, z_{n_k} = \alpha_{n_k}) = \\ &= \frac{1}{2} P(s_{n_1+d} = \alpha_{n_1}) P(s_{n_2} \oplus s_{n_2+d} = \alpha_{n_2}, \dots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k}) + \\ &\quad + \frac{1}{2} P(s_{n_1+d} = \bar{\alpha_{n_1}}) P(s_{n_2} \oplus s_{n_2+d} = \alpha_{n_2}, \dots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k}), \end{aligned}$$

to pa je po indukcijski predpostavki enako $(\frac{1}{2})^2(\frac{1}{2})^{k-1} + (\frac{1}{2})^2(\frac{1}{2})^{k-1}$, torej $(\frac{1}{2})^k$. Če pa se s_{n_1+d} v $P(s_{n_1+d} = \alpha_{n_1}, \dots, s_{n_k} \oplus s_{n_k+d} = \alpha_{n_k})$ pojavi še enkrat, upoštevamo

$$(s_{n_1+d} = \alpha_{n_1} \text{ in } s_{n_1+d} \oplus s_{n_1+2d} = \alpha_{n_1+d}) \iff (s_{n_1+d} = \alpha_{n_1} \text{ in } s_{n_1+2d} = \alpha_{n_1+d} \oplus \alpha_{n_1})$$

in potem računamo naprej kot smo začeli v tem odstavku. Pri $k = 1$ indukcijska predpostavka trivialno velja in s tem smo pokazali, da so spremenljivke $z_1, \dots, z_{n-d}$ med seboj neodvisne. Ker so tudi enako porazdeljene z

$$z_1 \sim \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \qquad \text{je} \qquad E(z_1) = \frac{1}{2} \quad \text{ter} \quad \mathrm{var}(z_1) = \frac{1}{4}$$

in lahko uporabimo centralni limitni izrek s katerim dobimo statistiko X_5. ■

Primer 4.1.1. *Vzemimo zaporedje s dolžine $n = 160$,*

$$\begin{aligned} s = \ &\texttt{11100 01100 01000 10100 11101 11100 10010 01001} \\ &\texttt{11100 01100 01000 10100 11101 11100 10010 01001} \\ &\texttt{11100 01100 01000 10100 11101 11100 10010 01001} \\ &\texttt{11100 01100 01000 10100 11101 11100 10010 01001} \end{aligned}$$

in ga testirajmo pri stopnji značilnosti $0'05$. Poker test naredimo za $m = 3$, torej štejemo pojavitve podzaporedij `000, 001, 010, 011, 100, 101, 110, 111` *na mestih $0, 3, 6, \dots, 156$. Podzaporedja se v zgoraj zapisanem vrstnem redu pojavijo $5, 10, 6, 4, 12, 3, 6$ in 7-krat. Za bločni test si najprej naračunamo števila e_i: $e_1 = 20'25$, $e_2 = 10'06$, $e_3 = 5'00$, $e_4 = 2'48$, ... Torej je $k = 3$. Preštejemo bloke in vrzeli: $B_1 = 25$, $B_2 = 4$, $B_3 = 5$, $V_1 = 8$, $V_2 = 20$ in $V_3 = 12$. Autokorelacijski test naredimo pri premiku $d = 3$.*

test	*interval zaupanja*	*vrednost X_i*	*rezultat testa*
monobit test	$[-1'9600, 1'9600]$	$0'6325$	$\surd$
test parov	$[0, 5'9915]$	$0'6252$	$\surd$
poker test	$[0, 14'0671]$	$9'6415$	$\surd$
bločni test	$[0, 9'4877]$	$31'7913$	//
autokorelacijski test	$[-1'9600, 1'9600]$	$-6'9434$	//

4.2 STANDARD FIPS 140-1

Ameriški standard FIPS[16] 140-1 določa štiri teste za preverjanje naključnosti generatorjev zaporedij bitov. Najprej generiramo zaporedje s dolžine 20000 bitov in preverimo, če so slučajne spremenljivke posameznih testov za s na predpisanih intervalih zaupanja.

[16]Federal Information Processing Standards

1. **Monobit test**. Število enic n_1 mora biti na intervalu $[9655, 10345]$.
2. **Poker test**. Slučajna spremenljivka X_3, definirana v enačbi (16), se računa za $m = 4$ in njena vrednost mora biti na intervalu $[1'03, 57'4]$.
3. **Bločni test**. Preštejemo vse bloke in vrzeli dolžine največ 6 (bloke in vrzeli daljše od 6 bitov štejemo zaporedoma kar s števcema B_6 in V_6). Nato preverimo, če B_i in V_i ležijo na danih intervalih:

dolžina bloka / vrzeli	interval zaupanja
1	2267 – 2733
2	1079 – 1421
3	502 – 748
4	223 – 402
5	90 – 223
6	90 – 223

4. **Test dolgih blokov**. Noben blok niti nobena vrzel ne smeta biti daljša od 33 bitov.

Če izhodno zaporedje generatorja ne ustreza kateremu od testov, pravimo, da generator ne ustreza standardu. Zato pa je interval zaupanja zelo velik. Za aplikacije, ki zahtevajo visoko stopnjo varnosti, standard zahteva, da se testi izvedejo ob vsakem zagonu (P)RBG. FIPS 140-1 dopušča, da se ti štirje testi lahko nadomestijo tudi s kakšnimi drugimi, ki so ekvivalentni tem ali pa strožji.

Za monobit in poker test lahko iz danih intervalov na preprost način določimo stopnjo značilnosti testa:

$$\alpha_1 = 1 - \int_0^{\frac{(10345-9655)^2}{20000}} f_{\chi^2(1)}(x)dx \qquad \text{in} \qquad \alpha_2 = 1 - \int_{1,03}^{57,4} f_{\chi^2(15)}(x)dx\,.$$

V obeh primerih je $\alpha = 0,000001$, kar pomeni, da je napaka prve vrste s temi testi zelo redka, zato pa pogosteje pride do napake II. vrste. Razloga za izbran tako majhen α sta najbrž ta, ker poženemo teste mnogokrat in na zelo kratkih sekvencah.

4.3 CRYPT-XS TESTI

Crypt-XS testi (oznaka S pomeni stream) so bili sprogramirani za tekoče šifrirne algoritme. Njihov opis je povzet po navodilih za uporabo [19]. Programski paket vsebuje 8 testov, od katerih prvih pet preverja osnovne lastnosti, ki bi jih moral imeti dober (P)RBG in so opisane tudi v prvem razdelku tega poglavja. Šesti test določi število različnih podzaporedij, ki se pojavijo, ko pregledujemo dano zaporedje od začetka do konca. Sedmi test preverja, ali ima naš algoritem lastnosti linearnosti (majhna linearna zahtevnost ipd), kot jih da LFSR generator, ki ga posredno uporabljajo mnogi dobri PRBG. Z njim dobimo zaželjeno dolgo periodo in lepe osnovne statistične lastnosti, vendar pa zaradi linearnosti ni kriptografsko varen. Osmi test pa določa, ali se posamezni vzorci v zaporedju večkrat ponovijo v kratkih razmikih. Posamezen test nam ponavadi vrne vrednost α, ki jo potem primerjamo s stopnjo značilnosti $\alpha_0 = 0'001$. Zaskrbljujoče so torej vrednosti α, ki so znotraj kritičnega območja.

Reference, na katere se sklicujejo avtorji Cript-XS testov, so naštete pri vsakem testu posebej. Uporabljali pa so tudi dve splošni referenci:

[BHAT 77] G. Bhattacharyya and R. Johnson, *Statistical Concepts and Methods*, John Wiley & Sons, 1977.

[FOLK 84] L. J. Folks, *Handbook of Statistics, Combination of Independent Tests*, Vol. 4, Elsevier Science Publishers, 1984, 113–121.

Imen testov ne bomo prevajali. Crypt-XS testi vsebuje naslednje teste:

1. **Frequency test**
   Test je že opisan pri osnovnih testih na strani 48, vendar tu vzamemo drugačno statistiko. Če z n_1 označimo število enic v zaporedju $s = s_1, \dots, s_n$, s p pa njihov relativni delež, tj. $p = n_1/n$, potem je statistika
   $$Z = 2\sqrt{n}\left(p - \frac{1}{2}\right)$$
   pri zaporedju, generiranim s pravim RBG, za dovolj velike n porazdeljena skoraj standardno normalno. Vrednost α določimo z
   $$\alpha = \frac{1}{\sqrt{2\pi}} \int_0^Z e^{-u^2/2} du\,,$$
   za kritično območje pa vzamemo unijo intervalov $[0, 0'0005]$ in $[0'9995, 0]$.

   Dokaz. Statistika Z je enaka vsoti $s_1 + \cdots + s_n$, zato lahko njeno limitno porazdelitev na preprost način dobimo iz centralnega limitnega izreka, kjer upoštevamo $\mu = E(s_1) = 1/2$ in $\sigma^2 = var(s_1) = 1/4$. Vrednost α nato določimo s pomočjo trditve (2.3.3). ■

2. **Binary derivative test**
   Test je zelo podoben osnovnemu testu parov s strani 49, preveri pa naslednji dve lastnosti, ki jih imajo zaporedja generirana s pravim RBG:

   (a) ali je število (simetričnih) parov `00` in `11` približno tako kot število parov `01` in `10` ter

   (b) ali je število (simetričnih) trojic `000`, `010`, `101` in `111` približno tako kot število trojic `100`, `001`, `110` in `011`.

   Ti dve lastnosti preverimo tako, da iz danega zaporedja $s = s_1, \dots, s_n$ izpeljemo novi zaporedji (derivative) $\mathrm{der}_1 = z_1, \dots, z_{n-1}$ in $\mathrm{der}_2 = w_1, \dots, w_{n-2}$, kjer je
   $$z_i = s_i \oplus s_{i+1} \qquad \text{in} \qquad w_i = (s_i \oplus s_{i+1}) \oplus (s_{i+1} \oplus s_{i+2})\,.$$
   V der_1 nam število ničel predstavlja število simetričnih podzaporedij dolžine 2, v der_2 pa število simetričnih podzaporedij dolžine 3. Na zaporedjih der_1 in der_2 zato naredimo monobit (frequency) test. Primer: V zaporedju `10100010000101` imamo torej pare `10`, `01`, `10`, `00`, `00`, `01`, `10`, `00`, `00`, `00`, `01`, `10` in `01` in iz njih dobimo der_1 = `1110011000111`. Podobno dobimo der_2 = `001010100100`.

   Referenca:

   [CARR 88] J. M. Carroll and L. E. Robbins, *Using binary derivatives to test an enhancement of DES*, Cryptologia, **12** (1988, No. 4), 193–208.

3. **Change point test**
   Test preveri ali ima zaporedje bitov $s = s_1, \dots, s_n$ do nekega delilnega mesta t približno toliko enic kot po delilnem mestu. Zaporedje s lahko ustreza monobit testu, a pade na

change point testu, če je npr. za $t = \lfloor \frac{1}{2}n \rfloor$ v zaporedju $s_1, \ldots, s_t$ delež enic enak $p_1 = \frac{1}{4}$, v zaporedju $s_{t+1}, \ldots, s_n$ pa je delež enic enak $p_2 = \frac{3}{4}$. Ocenjujemo torej vrednost $\max_{1 \le t \le n-1} |p_1 - p_2|$, kjer je p_1 relativni delež enic v zaporedju $s_1, \ldots, s_t$ in p_2 njihov relativni delež v zaporedju $s_{t+1}, \ldots, s_n$. Če je ta blizu 0, zaporedje ustreza testu, če pa je razlika prevelika, zaporedje ni dobro. Seveda ne smemo zavrniti zaporedja pri majhnih t-jih. Če je do s_{10} število enic 10, od s_{11} do $s_{500.000}$ pa jih je 250.000, to še ni razlog za zavrnitev hipoteze. Torej je treba vrednost $p_1 - p_2$ obtežiti z neko utežjo, ki je majhna pri t-jih na robovih intervala $[0, n]$, velika pa na sredini intervala $[0, n]$. V našem testu vzamemo utež $t(n-t)$. Definiramo funkcijo u,

$$u(t) = \underbrace{t(n-t)}_{\text{utež}} |p_1 - p_2| \,,$$

in poiščemo vrednost statistike

$$U = \max_{1 \le t \le n-1} u(t) \,.$$

Vrednost α potem aproksimiramo z

$$\alpha = e^{-2U^2/[n(p_1t + p_2n - p_2t)(n - p_1t - p_2n + p_2t)]}$$

in vzamemo za kritično območje interval $[0'999, 1]$.

Referenca:

[PETT 79] A. N. Pettitt, *A non-parametric approach to the change-point problem*, Appl. Statist., **28** (1979, No. 2), 126–135.

4. **Poker test**

Test je že opisan pri osnovnih testih na strani 50, program Crypt-XS pa ga izvede pri $m = 8$.

Referenca:

[BEKE 82] H. Beker and F. Piper, *Cipher Systems, The Protection of Communications*, Northwood Books, London, 1982.

5. **Runs test**

Test je že opisan pri osnovnih testih na strani 51.

Referenca:

[MOOD 40] A. M. Mood, *The distribution theory of runs*, Annals of Mathematical Statistics, **11** (1940), 367–392.

6. **Sequence complexity test**

S tem testom lahko nadomestimo osnovni autokorelacijski test, ki je opisan na strani 51, saj če zaporedje pade na autokorelacijskem testu, pade tudi na sequence complexity testu.

Test določi število c različnih podzaporedij v zaporedju $s = s_1, \ldots, s_n$, ko ga pregledujemo od začetka do konca. Algoritmični korak v računanju c bi tako opisali z besedami: "Za bitom s_j postavimo navpično črto, če smo zadnjo navpičnico postavili za bitom s_i in se zaporedje $s_{i+1}, \ldots, s_j$ ne pojavi kot podzaporedje nikjer v zaporedju $s_1, \ldots, s_{j-1}$." Ponazorimo to še na primeru, ko je

$$s = \texttt{1001111011000010} \,.$$

Postavimo navpičnico:

$$s = \texttt{1}|\texttt{0}|\texttt{01}|\texttt{1110}|\texttt{1100}|\texttt{0010}|$$

in število postavljenih navpičnic je vrednost c. V našem primeru je $c = 6$.

Porazdelitev slučajne spremenljivke c je opisana v [MUND 90]. V izpisu rezultatov testa je izpisana povprečna velikost te spremenljivke, ter mejna vrednost (treshold value) c_{tv} za c

$$c_{\text{tv}} = \frac{n}{\log_2 n}\,.$$

Test zavrne zaporedja, pri katerih je vrednost c v kritičnem območju, torej na intervalu $[c_{\text{tv}}, 1]$, z argumentom, da so vzorčasta (patterned).

Reference:

[DAWS 91] E. Dawson, *Design and Cryptanalysis of Symetric Ciphers, PhD Thesis*, QUT, 1991.

[LEMP 76] A. Lempel and J. Ziv, *On the complexity of finite sequences*, IEEE Trans. of Information Theory, **IT-22** (1976, Jan.), 75–81.

[MUND 90] S. Mund, *Ziv-Lempel Complexity for Periodic Sequences and its Cryptographic Application*, Proceedings of CRYPTO, 1990, 55–60.

7. **Linear complexity test**
Za razumevanje testa si je potrebno prebrati razdelek 1.5, ki nam predstavi LFSR generator psevdonaključnih števil. Dolžina najmanjšega LFSR generatorja, ki nam zgenerira zaporedje $s = s_1, \ldots, s_n$, se imenuje **linearna zahtevnost zaporedja** (linear complexity of the stream) in jo označimo z $L(s)$. V razdelku 1.5 je tudi razloženo, zakaj je LFSR generator tako zelo uporaben, kljub temu da ni kriptografsko varen. To pa je razlog za temeljito testiranje linearnosti v zaporedju s.

Primer 4.3.1.

$$s = \texttt{0101100101010010011110000011011100110001110101111101101}$$

Najmanjša linearna rekurzivna zveza, ki zgenerira to zaporedje, je

$$s_{t+6} = s_{t+5} \oplus s_{t+4} \oplus s_{t+1} \oplus s_t\,,$$

kjer je operacija $\oplus$ seštevanje po modulu 2, $x \oplus y = (x + y) \mod 2$. To pomeni, da je linearna zahtevnost zaporedja enaka $L(s) = 6$.

$L(s)$ je za zaporedje s, zgeneriranim s pravim RBG, slučajna spremenljivka porazdeljena po zakonu $N\left(\frac{1}{2}n, \frac{86}{81}\right)$, kar je dokazano v [RUEP 84] in [KREY 81]. Ker so zaporedja z veliko linearno zahtevnostjo povsem zaželjena, bomo pri testni funkciji

$$\alpha = \int_{-\infty}^{L(s)} f_{N\left(\frac{1}{2}n, \frac{86}{81}\right)}(t)dt$$

za kritično območje vzeli interval $[0, 0'001]$.

Problem pri rezultatu tega testa je, da test ne javi napake, če je zaporedje narejeno iz velikih vzorcev. Npr. pri zaporedju $s = \texttt{0000000100000001}$ je $L(s) = \frac{1}{2}n$, vendar napako javijo že prejšnji testi. Obstajajo pa zaporedja, ki so narejena iz velikih vzorcev in jih prejšnji testi ne zavrnejo, zato naredimo naslednja dva testa.

Pregled linearne zahtevnosti je zaporedje $L(s^k)$, $k = 1, \ldots, n$, kjer smo z s^k označili zaporedje $s_1, \ldots, s_k$. V [MASS 69] je pokazano, da naj bi bilo $L(s^k)$ za vsak k približno $\frac{1}{2}k$. Vsakemu indeksu k, $1 \le k \le n$, za katerega je $L(s^k) > L(s^{k-1})$, rečemo **skok** (jump).

Testiranje števila skokov (Test on the Number of "Jumps")
Število vseh skokov označimo z F. Za velike n je slučajna spremenljivka F porazdeljena po zakonu $N\left(\frac{1}{4}n, \frac{1}{8}n\right)$, kar je pokazano v [CART 87]. Zato je statistika

$$\frac{F - \frac{1}{4}n}{\sqrt{\frac{1}{8}n}}$$

potem porazdeljena standardno normalno. Ker so slaba zaporedja le tista s premalo skoki, bo kritično območje interval $[0, 0'001]$.

Testiranje porazdelitve skokov (Test on the Distribution of "Jumps")
Z f_h označimo število skokov linearne zahtevnosti z višino h. Višina skoka pri indeksu i je definirana kot $h_i = L(s_i) - L(s_{i-1})$. Verjetnost, da ima skok višino h, je $p_h = (\frac{1}{2})^h$. Matematično upanje za število skokov višine h je enako $E_h = p_h \cdot F$. Največjo višino skoka h, za katero je $E_h > 5$ označimo z m. Statistika

$$\chi^2 = \sum_{h=1}^{m} \frac{(f_h - E_h)^2}{E_h}$$

je pri zaporedju, generiranim s pravim RBG, porazdeljena po zakonu $\chi^2(m-1)$, glej [CART 87]. Za spremenljivko

$$\alpha = \int_0^{\chi^2} f_{\chi^2(m-1)}(t)dt$$

potem vzamemo kritično območje $[0'999, 1]$.

Reference:

[CART 87] G. Carter, *A statistical test for randomness based on the linear complexity profile of a binary sequence*, Technical Report for Racal Comsec Ltd., 1987.

[KREY 81] E. Kreysig, *Introductory Mathematical Statistics*, John Wiley and Sons, 1981.

[MASS 69] J. L. Massey, *Shift register sequences and BCH decoding*, IEEE Transactions on Information Theory, **IT-15** (1969, Jan.), 122–127.

[RUEP 84] R. A. Rueppel, *New Approaches to Stream Ciphers, PhD Thesis*, Swiss Federal Institute of Technology, 1984.

8. **Universal test**
V splošni uporabi so mnogi programi za arhiviranje (stiskanje, kompresiranje brez izgube) podatkov, kot npr. WinZip. Ti programi si dano zaporedje zapomnijo na malce drugačen način, kot je podano, pri tem pa ne završejo nobene informacije. Ta test poskusi kompresirati dano zaporedje ter izračuna dolžino novega zaporedja. Če je nova dolžina občutno krajša od prvotne, potem zaporedja ne bi smeli uporabljati za šifriranje.

Za testiranje si izberemo tri parametre: L, Q in K. Potem iz danega zaporedja bitov $s = s_1, \ldots, s_n$ naredimo neskončno zaporedje $z = z_1, z_2, \ldots$, tako da je $z_i = s_n$ za vsak

$i \geq 1$, ki je deljiv z n, in $z_i = s_{i \bmod n}$ za vse ostale $i \geq 1$. Zaporedje z nato razdelimo na L bitov dolga podzaporedja, ki se ne prekrivajo. Ta podzaporedja potem določajo člene novega zaporedja $w = w_1, w_2, \ldots$, kar pomeni, da je $w_i = z_{(i-1)L+1}, \ldots, z_{(i-1)L+L}$. Prvih Q členov imenujemo inicializacijski členi, na naslednjih K členih pa dejansko določimo, kako dobro se jih da kompresirati. Najprej definiramo funkcijo Tab, ki nam pri vsakem indeksu i pove, pri katerem indeksu, se je člen w_i nazadnje pojavil v zaporedju w, torej je

$$\mathrm{Tab}(i) = \begin{cases} 0 & ;\ w_k \neq w_i \text{ za vsak } k,\ 1 \leq k \leq i-1 \\ \max\{k;\ w_k = w_i,\ k \leq i-1\} & ;\ \text{sicer} \end{cases} .$$

Sedaj lahko določimo vrednost statistike

$$f = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log_2[i - \mathrm{Tab}(i)]$$

in nato še vrednost slučajne spremenljivke

$$z = \frac{f - E(f)}{c(L,K)\sqrt{\mathrm{var}(f)}} ,$$

ki je ne moremo dobiti direktno iz centralnega limitnega izreka, saj sumandi v statistiki f niso povsem neodvisni, zato je tudi prisoten korekcijski faktor

$$c(L,K) \approx 0'7 - 0'8\frac{1}{L} + (1'6 + 12'8\frac{1}{L})K^{-\frac{4}{L}} .$$

Slučajna spremenljivka z je standardno normalno porazdeljena, zato določimo vrednost α iz

$$\alpha = \int_{-\infty}^{z} f_{N(0,1)}(t)dt ,$$

za kritično območje statistike α pa vzamemo interval $[0, 0'001]$. Pri izračunu smo potrebovali količini

$$E(f) = \sum_{i=1}^{\infty} \frac{1}{2^L}\left(\frac{2^L-1}{2^L}\right)^{i-1} \log_2 i \qquad \text{in}$$
$$\mathrm{var}(f) = \sum_{i=1}^{\infty} \frac{1}{2^L}\left(\frac{2^L-1}{2^L}\right)^{i-1} (\log_2 i)^2 - \left(\sum_{i=1}^{\infty} \frac{1}{2^L}\left(\frac{2^L-1}{2^L}\right)^{i-1} \log_2 i\right)^2 .$$

Primer 4.3.2. *Test naredimo na zaporedju*

$$s = \mathtt{1,0,1,0,0,0,1,0,0,0,0,1,0,1,1,1,0,0,0,1,0,1,1,0,0,} \\ \mathtt{0,1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,0,0,0,1} \quad ,$$

kjer vzamemo $L = 3$, $Q = 16$ in $K = 17$. Da bo zapis bolj pregleden, upoštevajmo $L = 3$, izločimo vejice iz zapisa zaporedja s in pišimo po 3 bite skupaj,

$$s = \mathtt{101|000|100|001|011|100|010|110|001|110|101|010|101|010|100|000|01.}$$

Tvorimo še zaporedje z, tako da zaporedje s periodično nadaljujemo,

$$z = \mathtt{101|000|100|001|011|100|010|110|001|110|101|010|101|010|100|000|011|} \\ \mathtt{010|001|000|010|111|000|101|100|011|101|010|101|010|101|000|000|1\ldots.}$$

Zaporedje w si lahko predstavljamo, tako da tri zaporedne bite $z_i z_{i+1} z_{i+2}$ spremenimo v število $4z_i + 2z_{i+1} + z_{i+2}$,

$$w = 5,0,4,1,3,4,2,6,1,6,5,2,5,2,4,0,3,2,1,0,2,7,0,5,4,\\ 3,5,2,5,2,5,0,0,1,\dots .$$

Sedaj določimo še zaporedje $\mathrm{Tab}(i)$,

$$\mathrm{Tab} = 0,0,0,0,0,3,0,0,4,8,1,7,11,12,6,2,5,14,9,16,18,0,20,13,15,17,\\ 24,21,27,28,29,23,32,10,25,\dots ,$$

in potem še vrednost statistike f, ki je enaka $f = 2.29705$. Iz nje potem določimo še statistiko $z = -0'133611$, kjer smo upoštevali $c(3,17) = 0'567546$. Vrednost α je potem enaka $\alpha = 0'446855$ in test ne zavrne zaporedja.

Referenca:

[MAUR 92] U. M. MAURER, *A Universal Statistical Test for Random Bit Generators*, Journal of Cryptology, **5** (1992), 89–105.

4.4 DIEHARD TESTI

Dr. George Marsaglia je profesor na oddelku za statistiko na Ameriški univerzi Florida State University in aktivno raziskuje tudi generatorje naključnosti na Supercomputer Computations Research Institute. O generatorjih naključnosti je objavil že mnogo člankov (dostopnih tudi na internetu, npr. [13] in [14]) v uglednih revijah, razvil pa je tudi statistične teste za preverjanje naključnosti pri (P)RBG, glej [20] in [21]. To so Diehard testi, ki so dostopni tudi na internetu, o njihovi množični uporabi pa govori tudi število zadetkov na spletnih iskalnikih, kjer je razvidno, da je na strani, kjer so na voljo program Diehard in omenjeni članki, usmerjenih mnogo kazalcev z vseh koncev sveta. O uporabnosti raziskovalnih dosežkov G. Marsaglie je zelo zgovoren tudi podatek o trinajstih referencah v Knuthovem priročniku [11].

Diehard testi ne preverjajo osnovnih lastnosti, ampak bolj "neobičajne", nekatere so posredno povezane z zgradbo mnogih generatorjev. Diehard program vrši 17 testov, razvitih za potrebe testiranja generatorjev naključnosti, ki se uporabljajo pri igrah na srečo. Za testiranje potrebujemo 10 Mbytov[17] veliko datoteko, generirano z (P)RBG. Opis testov je vzet iz datoteke rezultatov testiranja ter člankov [13] in [14].

Večina Diehard testov vrne vrednost spremenljivke p, ki naj bi bila slučajna in enakomerno porazdeljena na intervalu $[0,1)$. To je pravzaprav statistična domneva, ki jo preverjamo. Ti p-ji so dobljeni iz porazdelitve statistike X, ki jo računamo v posameznem testu, kot je opisano v trditvi (2.3.3). Za porazdelitev X naredimo aproksimacijo z neko realno slučajno spremenljivko. Ker je X ponavadi omejena, bo ta aproksimacija najslabša za velike vrednosti. Tudi ne smemo biti presenečeni, če je kakšna vrednost p blizu 0 ali 1 (npr. 0'0012 ali pa 0'9983), saj je to povsem normalno ob stotinah testov, ki jih naredi program Diehard. Če je niz res slab, bo kar nekaj (šest ali več) p-jev enakih 0 ali 1 (zaokroženo na 6 decimalk).

Diehard testi vsebujejo naslednje teste:

1. **Birthday spacings test**
   Naključno izberemo m rojstnih dni in jih označimo z $I_1, I_2 \dots, I_m$ (poljuben dan izberemo

[17] 1 byte = 8 bitov, 1 Mbyte = 1024^2 bytov = 1.048.576 bytov

z verjetnostjo $\frac{1}{m}$) v letu, ki ima n dni. Nato I-je uredimo naraščajoče po velikosti in tako dobljena števila po vrsti označimo z $I_{(1)}, I_{(2)} \ldots, I_{(m)}$. Nato določimo dolžine presledkov med posameznimi praznovanji

$$I_{(1)}, I_{(2)} - I_{(1)}, I_{(3)} - I_{(2)}, \ldots, I_{(m)} - I_{(m-1)} .$$

Če je

$$j = m - \text{“število vrednosti, ki se v zadnjem zaporedju pojavijo natanko enkrat”} ,$$

potem se porazdelitev spremenljivke j asimptotično približuje Poissonovi porazdelitvi $Po(\frac{m^3}{4n})$, kar je dokazal Janos Komlos. Izkušnje kažejo, da mora biti n dovolj velik, da je ta aproksimacija dobra, recimo $n \geq 2^{18} \approx 2'6 \cdot 10^5$. Ta test uporablja $n = 2^{24} = 1'7 \cdot 10^7$ in $m = 2^9 = 512$, tako da je $j \sim Po(2)$. Na tak način naračunamo 500 j-jev in to je vzorec za χ^2-test, katerega postopek je opisan na strani 43. Za test uporabimo particijo $E_1 = \{0\}$, $E_2 = \{1\}, \ldots, E_6 = \{5\}$ in $E_7 = \{6, 7, \ldots\}$. Verjetnost, da ima spremenljivka j vrednost v množici E_i, določimo iz $Po(2)$ porazdelitve. Te verjetnosti so $p_1 = \frac{2^0}{0!}e^{-2} \doteq 0'13534$, $p_2 = \frac{2^1}{1!}e^{-2} \doteq 0'27067$, $p_3 = \frac{2^2}{2!}e^{-2} \doteq 0'27067$, $p_4 = \frac{2^3}{3!}e^{-2} \doteq 0'18045$, $p_5 = \frac{2^4}{4!}e^{-2} \doteq 0'09022$, $p_6 = \frac{2^5}{5!}e^{-2} \doteq 0'03609$, $p_7 = 1 - \sum_{i=1}^{6} p_i \doteq 0'01656$, kar pomeni, da so pričakovane vrednosti za N_i enake $500p_i$ oz. $67'668$, $135'34$, $135'34$, $90'224$, $45'112$, $18'045$, $8'2818$ zaporedoma. Iz teh vrednosti določimo vrednost spremenljivke $Y = \sum_{i=1}^{7} \frac{(N_i - 500p_i)^2}{500p_i}$ ter vrednost p, ki jo interpretiramo, kot piše v opisu postopka testa. Na tak način izračunamo 9 p-jev, ki vsak zase predstavljajo nek delen rezultat testa. Nato pa še na teh p-jih, ki naj bi bili enakomerno porazdeljeni, naredimo s KS testom končen rezultat Birthday testa. Za ta test se uporabi tisto različico testa Kolmogorova, ki dopušča, da vzorec ni povsem slučajen. Izreka v tem delu nismo nikjer navedli, saj je dokaz še težji od običajnega izreka Kolmogorova.

Za dejansko izvedbo testa na testni datoteki, sestavljeni iz bitov $b_1, b_2, \ldots$, pa je seveda potrebno določiti, kateri biti bodo določali posamezne rojstne dneve. Za prvi p (od devetih) nam bodo rojstne dneve predstavljala števila $b_{32k+1}b_{32k+2}\ldots b_{32k+24}$, torej $b_1 b_2 \ldots b_{24}$,-$b_{33}b_{34}\ldots b_{56}, \ldots$ Za vsakega od 500 j-jev potrebujemo $m = 512$ števil. Za drugi p vzamemo bite $b_{32k+2}b_{32k+3}\ldots b_{32k+25}$ in tako naprej do devetega p-ja, kjer potrebujemo bite $b_{32k+9}b_{32k+10}\ldots b_{32k+32}, k = 0, 1, ..., 500 \cdot 512$. Zatorej potrebujemo datoteko veliko 8.192.000 bitov (0'98 Mbytov).

2. **Overlapping 5-permutation test**
   Za test potrebujemo datoteko 1.000.000 32-bitnih naključnih celih števil $u_1, u_2, \ldots$. V testiranju tvorimo zaporedje peteric

$$(u_1, u_2, u_3, u_4, u_5),\ (u_2, u_3, u_4, u_5, u_6),\ (u_3, u_4, u_5, u_6, u_7), \ldots$$

   Vsaka od teh peteric nam predstavlja neko stanje S na sledeč način:

$$S\big((x_1, x_2, x_3, x_4, x_5)\big) = \begin{cases} 1 & ;\ x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \\ 2 & ;\ x_1 \leq x_2 \leq x_3 \leq x_5 < x_4 \\ 3 & ;\ x_1 \leq x_2 \leq x_4 < x_3 \leq x_5 \\ & \quad\vdots \\ 120 & ;\ x_5 < x_4 < x_3 < x_2 < x_1 \end{cases}$$

   Vsaka od zgornjih peteric je odvisna od svoje predhodnice (od nje prevzame 4 vrednosti od petih). Sedaj lahko rečemo, da vsako od števil u_i, $i = 5, 6, 7, \ldots$, določi neko stanje

$S\big((u_i, u_{i-1}, u_{i-2}, u_{i-3}, u_{i-4})\big)$ verige V. Tako iz zaporedja $u_5, u_6, \ldots$ dobimo zaporedje stanj. Z w_i označimo število pojavitev stanja i v zaporedju stanj, s C pa kovariančno matriko slučajnega vektorja $[w_i]_{i=1}^{120} \in \mathbb{Z}^{120}$. Statistiko X lahko poiščemo s pomočjo izreka (2.6.6) in je enaka

$$X = \sum_{i,j=1}^{120} (w_i - \mu_i) c^-_{ij} (w_j - \mu_j)\,,$$

njena porazdelitev pa se asimptotično približuje $\chi^2(99)$ porazdelitvi. Z μ_{ijk} smo označili matematično upanje vektorja $[w_{i,j,k}]$, s C^- pa katerikoli posplošen inverz matrike C, to je matrika C^-, za katero velja $CC^-C = C$. Če pravi inverz C^{-1} obstaja, zanj gotovo velja $CC^{-1}C = C$. Iskanje tega inverza za tako veliko matriko zahteva zelo podrobno znanje linearne algebre in z njo povezanih numeričnih metod. Nato iz vrednosti X le še določimo vrednost p. Ker bomo test izvedli dvakrat, potrebujemo datoteko veliko 8 Mbytov.

3. **Binary rank test for 31x31 matrices**
   Najbolj levih 31 bitov od 31 celih števil (32-bitnih) iz testne datoteke uporabimo za matriko A nad obsegom $\{0, 1\}$. Na tak način tvorimo 40.000 matrik in vsaki določimo rang[18], ki je med 0 in 31, vendar pa so rangi manjši od 28 redki in zato za χ^2-test na vzorcu rangov vzamemo particijo množice $\mathbb{Z}_{32}$: $E_1 = \{0, 1, 2, \ldots, 28\}, E_2 = \{29\}, E_3 = \{30\}$ in $E_4 = \{31\}$. Nato določimo vrednost p. Za test potrebujemo 4'73 Mbytov veliko datoteko.

4. **Binary rank test for 32x32 matrices**
   Vseh 32 bitov od 32 celih števil iz testne datoteke uporabimo za matriko A nad obsegom $0, 1$. Na tak način tvorimo 40.000 matrik in vsaki določimo rang, ki je med 0 in 32, vendar pa so rangi manjši od 29 redki in zato za χ^2-test na vzorcu rangov vzamemo particijo množice $\mathbb{Z}_{33}$: $E_1 = \{0, 1, 2, \ldots, 29\}, E_2 = \{30\}, E_3 = \{31\}$ in $E_4 = \{32\}$. Nato določimo vrednost p. Za test potrebujemo 4'88 Mbytov veliko datoteko.

5. **Binary rank test for 6x8 matrices**
   Najbolj levih 8 bitov od 6 celih števil (32-bitnih) iz testne datoteke uporabimo za matriko A nad obsegom $0, 1$. Na tak način tvorimo 100.000 matrik in vsaki določimo rang, ki je med 0 in 6, vendar pa so rangi manjši od 4 redki in zato za χ^2-test na vzorcu rangov vzamemo particijo množice $\mathbb{Z}_7$: $E_1 = \{0, 1, 2, 3, 4\}, E_2 = \{5\}$ in $E_3 = \{6\}$. Nato določimo vrednost p. Test ponovimo 25-krat, najprej smo to naredili na bitih 1–8 v izbranem 32 bitnem številu, sedaj pa to ponovimo še na bitih 2–9, 3–10, ..., 25–32. Na tako dobljenih 25 p-jih naredimo še KS test (različico, ki ne zahteva slučajnega vzorca).

   Za test potrebujemo 2'29 Mbytov veliko datoteko.

6. **Overlapping 20-tuples bitstream test**
   Iz testne datoteke vzamemo $2^{21} + 19 = 2097171$ bitov (256 kbytov) $b_1, b_2, ..., b_{2^{21}+19}$. Zamislimo si abecedo z dvema črkama, `0` in `1`. Potem iz testne datoteke sestavimo 20 črk dolge besede $b_1 b_2 \ldots b_{20}$, $b_2 b_3 \ldots b_{21}$, ... Tako dobimo 2^{21} besed. Vseh možnih 20 črk dolgih besed je 2^{20}. Z j označimo število tistih, ki manjkajo. Spremenljivka j naj bi bila porazdeljena $N(141909, 428^2)$, torej $\frac{j-1419009}{428} \sim N(0,1)$. Vrednost p tega testa dobimo z enačbo iz trditve 2.3.3, torej

$$p = \int_{-\infty}^{\frac{j-1419009}{428}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}\, dx\,.$$

   Test ponovimo 20-krat.

[18] maksimalno število linearno neodvisnih vrstic

Za test potrebujemo 5'00 Mbytov veliko datoteko.

7. **Overlapping-pairs-sparse-occupancy (OPSO)**
Najbolj levih 10 bitov od $2^{21}+1$ celih števil (32-bitnih) iz testne datoteke uporabimo za tvorjenje črke iz abecede z $2^{10} = 1024$ črkami. Na tem zaporedju črk $C_1C_2 \ldots C_{2^{21}+1}$ tvorimo besede, ki se prekrivajo (torej med sabo niso neodvisne) $B_1 = C_1C_2, B_2 = C_2C_3, B_3 = C_3C_4, \ldots, B_{2^{21}} = C_{2^{21}}C_{2^{21}+1}$. Vseh možnih besed sestavljenih iz dveh črk je $(2^{10})^2 = 2^{20}$ in če z j označimo število manjkajočih dvočrkovnih besed v zaporedju $B_1, B_2, \ldots B_{2^{21}}$, potem je
$$\frac{j - 141909}{290} \sim N(0,1)$$
Tu sta in matematično upanje in varianca dobljena z izračunom in ne, tako kot v nekaj naslednjih testih, s simulacijo. Za ta j nato, kot do sedaj, določimo vrednost p. Test ponovimo 23-krat. Najprej to naredimo na bitih 1–10 v izbranem 32 bitnem številu, nato pa to ponovimo še na bitih 2–11, 3–12, ... , 23–32.

Za test potrebujemo 8'00 Mbytov veliko datoteko.

8. **Overlapping-quadruples-sparse-occupancy (OQSO)**
Test je podoben testu OPSO, le da delamo z abecedo iz $2^5 = 32$ črk in tvorimo 4-črkovne besede. Z j označimo število manjkajočih 4-črkovnih besed v zaporedju $B_1, B_2, \ldots B_{2^{21}}$, kjer je $B_1 = C_1C_2C_3C_4, B_2 = C_2C_3C_4C_5, \ldots$. Za pravi RBG potem velja
$$\frac{j - 141909}{295} \sim N(0,1)\,.$$
Kot je navedeno v izpisu Diehard testa, je matematično upanje (141909) spremenljivke j dobljeno s teorijo, varianca (295^2) pa z izčrpno simulacijo. Za ta j nato, kot do sedaj, določimo vrednost p. Test ponovimo 28-krat, najprej smo to naredili na bitih 1–5 v izbranem 32 bitnem številu, sedaj pa to ponovimo še na bitih 2–6, 3–7, ... , 28–32.

Za test potrebujemo 8'00 Mbytov veliko datoteko.

9. **DNA test**
Test je zasnovan podobno kot testa OPSO in OQSO, le da tu vzamemo abecedo iz $2^2 = 4$ črk, ki jih označimo s `C`, `G`, `A` in `T`, ter tvorimo 10-črkovne besede. Z j označimo število manjkajočih besed v zaporedju $B_1, B_2, \ldots B_{2^{21}}$, kjer je $B_1 = C_1 \ldots C_{10}, B_2 = C_2 \ldots C_{11}, \ldots$. Za pravi RBG potem velja
$$\frac{j - 141909}{339} \sim N(0,1)\,.$$
Tudi tu je varianca (339^2) dobljena s simulacijo. Za ta j nato, kot do sedaj, določimo vrednost p. Test ponovimo 31-krat. Najprej smo to naredili na bitih b_1b_2 v izbranem 32 bitnem številu, sedaj pa to ponovimo še na bitih b_2b_3, b_3b_4, ... , $b_{31}b_{32}$.

Za test potrebujemo 8'00 Mbytov veliko datoteko.

10. (a) **Count-the-1's test na zaporedju bytov**
Testno datoteko si mislimo kot zaporedje bytov (8 bitov). Vsak byte lahko vsebuje od 0 pa do 8 enic z verjetnostmi $\frac{1}{256}$, $\frac{8}{256}$, $\frac{28}{256}$, $\frac{56}{256}$, $\frac{70}{256}$, $\frac{56}{256}$, $\frac{28}{256}$, $\frac{8}{256}$ in $\frac{1}{256}$. Sedaj vsak byte spremenimo v črko `A`, `B`, `C`, `D` ali `E`, kot je prikazano v tabeli.

število enic v bytu	ustrezna črka
0,1 ali 2	`A`
3	`B`
4	`C`
5	`D`
6,7 ali 8	`E`

To je tako kot bi tipkali črke `A`, `B`, `C`, `D` in `E` z verjetnostmi $\frac{37}{256}$, $\frac{56}{256}$, $\frac{70}{256}$, $\frac{56}{256}$ in $\frac{37}{256}$. Na razpolago imamo 5^5 možnih 5-črkovnih besed. Podobno kot v prejšnjih testih tu naredimo iz 256000 petčrkovnih besed, ki se prekrivajo (na nek način imamo zopet Markovsko verigo[19], kot v overlapping 5-permutation testu). Test je po zgradbi podoben prejšnjim štirim, le da je tu statistična obdelava drugačna. Prešteli bomo pojavitve posameznih besed in iz posplošenega inverza izračunane kovariančne matrike C za štetje besed bomo določili kvadratno formo, kar bo osnova za χ^2-test, ki ga bomo naredili malo drugače. Najprej bomo določili Pearsonovo χ^2-statistiko Y_5 za kvadratne forme, nato pa bomo test naredili še za 4-črkovne besede in določili še Pearsonovo χ^2-statistiko Y_4. Spremenljivka $Y_5 - Y_4$ je (ne bomo dokazovali) porazdeljena χ^2 s $5^5 - 5^4$ prostorskimi stopnjami, torej

$$Y_5 - Y_4 \sim \chi^2(2500)\,.$$

Test ponovimo dvakrat. Za test potrebujemo 0'98 Mbytov veliko datoteko.

(b) **Count-the-1's test na točno določenih bytih**
Tu tvorimo iz datoteke zaporedje bitov, tako da gledamo na datoteko kot na zaporedje 32-bitnih celih števil in iz vsakega števila vzamemo 8 točno določenih zaporednih bitov. Na tako dobljenem zaporedju naredimo zgoraj opisani Count-the-1's test. Test naredimo 25-krat, na bitih 1–8, 2–9, 3–10, ... , 25–32.
Za test potrebujemo 3'91 Mbytov veliko datoteko.

11. **Parking lot test**
Na parkirišču, kvadratu s stranico dolgo 100, parkiramo avtomobile, kroge z radijem 1, "po posluhu". To pomeni, da si naključno in neodvisno od ostalih izborov izberemo neko mesto in tam poskusimo parkirati avto. Če se zaletimo v drug avto, nič zato, naključno izberemo novo parkirišče, vse dokler nam ne uspe. Z n označimo število poskusov parkiranja, s k pa število uspešno parkiranih avtomobilov. V testu vzamemo $n = 12.000$. S simulacijo je bilo ugotovljeno, da se porazdelitev

$$\frac{k - 3523}{21'9}$$

približuje standardni normalni porazdelitvi. Za naš test določimo 10 k-jev in za vsakega vrednost p. Test ponovimo 10-krat in iz tako dobljenih 10 p-jev naredimo še KS test.

12. **Minimum distance test**
V kvadratu s stranico 10.000 naključno izberemo $n = 8.000$ med seboj neodvisnih točk in nato poiščemo najkrajšo razdaljo d med dvema točkama.

$$d = \min_{\substack{1<i<10.000\\1<j<10.000}} d(T_i, T_j)$$

Spremenljivka d^2 naj bi bila eksponentno porazdeljena s parametrom 0'995, kar pomeni, da je

$$p = 1 - e^{-\frac{1}{0'995}d^2} \sim I[0,1]\,.$$

[19] Zaporedje slučajnih spremenljivk, ki izpolnjuje predpisane lastnosti. Natančna definicija in pomembne lastnosti so opisane npr. v učbeniku verjetnostnega računa [6].

Test ponovimo 100-krat in na tako dobljenih 100 p-jih naredimo še KS test.

13. **3D-spheres test**
V kocki s stranico 1.000 naključno izberemo $n = 4.000$ med seboj neodvisnih točk in nato poiščemo najkrajšo razdaljo d med dvema točkama.

$$d = \min_{\substack{1<i<10.000 \\ 1<j<10.000}} d(T_i, T_j)$$

Spremenljivka d^3 naj bi bila eksponentno porazdeljena s parametrom 30. Le ta je dobljen s simulacijo. To pomeni, da je

$$p = 1 - e^{-\frac{1}{30}d^3} \sim I[0,1]\,.$$

Test ponovimo 20-krat in na tako dobljenih 20 p-jih naredimo še KS test.

14. **Sqeeze test**
Na datoteko gledamo kot na zaporedje celih števil (4 byti). Vsakega od teh števil pretvorimo v realno število U (npr.
$45 = \texttt{00000000\,00000000\,00000000\,00101101}_{(2)}$
$\to 0{,}00000000\,00000000\,00000000\,00101101_{(2)}$
$\to 1.0477378964 \cdot 10^{-8}$
Posamezen test določi število iteracij j potrebnih, da reduciramo začetni $k = 2^{31} = 2.147.483.648$ na 1 z redukcijo $k = \lceil k \cdot U \rceil$. Test poišče 100.000 j-jev, prešteje posamezne frekvence j-jev v množicah $E_1 = \{0, 1, 2, \ldots, 6\}$, $E_2 = \{7\}$, $E_3 = \{8\}$, ... , $E_{42} = \{47\}$ in $E_{43} = \{48, 49, \ldots\}$ in naredi na njih χ^2-test ter določi vrednost p. Za test potrebujemo 0'38 Mbytov veliko datoteko.

15. **Overlapping sums test**
Tudi tu iz 4-bytnih celih števil naredimo realna števila $U_1, U_2, \ldots$. Le ta naj bi bila enakomerno porazdeljena na $[0, 1)$. Sestavimo (med seboj odvisne) vsote $S_1 = U_1 + \ldots + U_{100}$, $S_2 = U_2 + \ldots + U_{101}$, ... Porazdelitev S-jev je asimptotično večrazsežna normalna z neko kovariančno matriko (ki ni diagonalna, saj so S-ji med seboj odvisni). Nato z linearno transformacijo S-jev naredimo zaporedje med seboj neodvisnih standardnih normalnih spremenljivk, ki jih nato z enačbo iz trditve (2.3.3) pretvorimo v enakomerno porazdeljene q-je na katerih naredimo KS test in dobimo vrednost p. Test ponovimo 10-krat in iz tako dobljenih desetih p-jev dobimo s še enim KS testom končno vrednost p.

16. **Runs test**
Tu iz 10.000 4-bytnih celih števil naredimo realna števila U_1, U_2, ... , $U_{10.000}$. Med njimi postavimo neenačaje $\geq$ in $<$ in z njihovo pomočjo tvorimo zaporedje bitov, kot prikazuje primer:

```
   0'123 < 0'357 < 0'789 > 0'425 > 0'224 < 0'416 < 0'950 ...
 1         1       1       0       0       1       1   ...
```

V tem zaporedju preštejemo dolžine blokov in vrzeli:

```
         3                        2                      ?       ...
```

In tvorimo novo zaporedje (runs-up and runs-down) (3,2,?, ...). Kovariančna matrika za tako zaporedje slučajnih spremenljivk je znana. Sedaj naredimo le še test na kvadratni formi s posplošenim inverzom kovariančne matrike, glej izrek (2.6.6). Test ponovimo dvakrat.

Za test potrebujemo 0'076 Mbytov veliko datoteko.

17. **Craps test**
Test igra 200.000 iger Craps[20], kjer ena igra traja, dokler se ne poračuna stava na pass line. Test poišče število zmag ter število metov, potrebnih za končanje posamezne igre.

Število zmag Z se asimptotično približuje normalni porazdelitvi s povprečjem $200.000q$ in varianco $\sigma^2 = 200.000q(1-q)$, kjer je $q = \frac{244}{495}$. Vrednost p dobimo s pomočjo trditve (2.3.3), $p = \int_{-\infty}^{Z} f_{N(200.000q,(200.000q(1-q))^2)}(x)dx$.

Število metov, potrebnih za končanje igre, pa je lahko kakršnokoli, vendar pa se redko zgodi, da več kot 21. Zato za χ^2-test, ki ga naredimo na številu metov, vzamemo naslednjo particijo naravnih števil: $E_1 = \{1\}$, $E_2 = \{2\}$, $E_3 = \{3\}$, ... , $E_{20} = \{20\}$ in $E_{21} = \{21, 22, 23, \dots\}$. Ustrezne verjetnosti so $p_1 \doteq 0'333333$, $p_2 \doteq 0'188272$, $p_3 \doteq 0'134774$, $p_4 \doteq 0'0965673$, $p_5 \doteq 0'0692571$, $p_6 \doteq 0'0497177$, $p_7 \doteq 0'0357252$, $p_8 \doteq 0'0256954$, $p_9 \doteq 0'0184993$, $p_{10} \doteq 0'0133315$, $p_{11} \doteq 0'00961665$, $p_{12} \doteq 0'00694370$, $p_{13} \doteq 0'00501858$, $p_{14} \doteq 0'00363070$, $p_{15} \doteq 0'00262918$, $p_{16} \doteq 0'00190575$, $p_{17} \doteq 0'00138270$, $p_{18} \doteq 0'00100415$, $p_{19} \doteq 0'000729922$, $p_{20} \doteq 0'000531076$ in $p_{21} \doteq 0'00143557$. Vsako 32-bitno celo število z v testni datoteki pretvorimo v met kocke tako da ga najprej spremenimo v realno število na način, ki je opisan že v prejšnjih testih, dobljeno množimo s 6, vzamemo celi del in prištejemo 1.

Za test v povprečju potrebujemo 5'15 Mbytov, saj ena igra traja v povprečju $\frac{557}{165}$ metov, za en met pa potrebujemo 8 bytov.

[20]Igra **Craps** je ameriška igra na srečo, ki se igra v casinojih in jo dostikrat vidimo v filmih. Igralec meče proti "steni" dve kocki. Met je veljaven, če se kocki odbijeta od stene, rezultat meta pa je vsota pik na kockah. Prvi met se imenuje **come out roll**. Igra traja, dokler igralec ne dobi, oz. zgubi stave na **pass line**. Igralec lahko zmaga (**win**) na pass line na dva načina. Prvi način je, da je rezultat prvega meta 7 ali 11. Drugi način pa, ko je prvi met 4, 5, 6, 8, 9 ali 10, postane to metalčeva **ključna številka** (**point**). Da bi zmagal na pass line, jo mora v naslednjih metih ponoviti predno vrže sedmico. Igralec zgubi stavo na pass line, če je rezultat prvega meta **craps**, to je 2, 3 ali 12, ali če doseže ključno številko in jo ne uspe ponoviti pred sedmico. Obstaja še dosti drugih stav, vendar pa je to za opis testa dovolj.

Slike

Tabele

Literatura

[1] T. W. Anderson, D. A. Darling, *Asimptotic theory of certain “goodness of fit” criteria based on stohastic processes*, Annals of Mathematical Statistics, **23** (1952), 193–212.

[2] H. Beker, F. Piper, *Cypher Systems: The Protection of Comunications*, Northwood Publications, London, 1982.

[3] P. Billingsley, *Probability and measure, 2nd ed.*, John Wiley, New York, 1986.

[4] R. Durrett, *Probability: theory and examples, 2nd ed.*, Duxbury Press, Belmont, 1996.

[5] M. Fisz, *Probability Theory and Mathematical Statistics*, PWN, Polish Scientific Publishers, New York, 1967.

[6] G. R. Grimmett, D. R. Stirzaker, *Probability and Random Processes*, Clarendon Press, Oxford, 1992.

[7] P. Jakopin, *Zgornja meja entropije pri leposlovnih besedilih v slovenskem jeziku: doktorska disertacija*, samozaložba, Ljubljana, 1999.
`http://valjhun.fmf.uni-lj.si/~ajurisic/tecaj1/entropija.txt`

[8] R. Jamnik, *Matematična statistika*, Državna založba Slovenije, Ljubljana, 1980.

[9] R. Jamnik, *Verjetnostni račun*, Mladinska knjiga, Ljubljana, 1971.

[10] J. F. C. Kingman, S. J. Taylor, *Introduction to measure and probability*, Cambridge University Press, Cambridge, 1966.

[11] D. E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, Addison-Wesley Publishing Company, 1969. B. Košmelj, *Statistični terminološki slovar*, Statistično društvo Slovenije: Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana, 1993.

[12] Susan Landau, *Communications Security for the Twenty-first Century: The Advanced Encryption Standard*, Notices of the AMS, **47** (2000), 450–459.

[13] G. Marsaglia, *A Current View of Random Number Generators*, Proceedings of the Sixteenth Symposium on the Interface (Atlanta, Georgia, March 1984), Computer Science and Statistics, Elsevier Science Publishers, New York, 1985, str. 3–10.
`http://www.evensen.org/marsaglia/`
Pseudo random number generators and stringent tests for randomness (keynote.ps)

[14] G. Marsaglia, *Monkey Tests for Random Number Generators*, Computers & Mathematics with Applications, **9** (1993), 1–10.
`http://www.evensen.org/marsaglia/`

[15] A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press LLC, 1997.

[16] W. Rudin, *Real and complex analysis, 3rd ed.*, McGraw-Hill, New York, 1987.

[17] M. J. Schervish, *Theory of Statistics*, Springer-Verlag, New York, 1995.

[18] D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.

Internet, navodila za uporabo programov, ...

[19] Navodila za uporabo Crypt-XS testa:
W. Caelli, E. Dawson, H. Gustafson, L. Nielsen, *Crypt - XS, Statistical package manual for stream ciphers*, Queensland university of technology, Information security research centre and School of mathematics, 1994.
`http://www.isrc.qut.edu.au/cryptx/`

[20] Razlage Diehard testov, ki jih izpiše program:
`http://www.stata.com/support/cert/diehard/randnumb.out`

[21] Programski paket Diehard:
`http://stat.fsu.edu/~geo/diehard.html`

Stvarno kazalo