Δημήτριος Κυριακίδης
el17077

Κρυπτογραφία
1η Φυλλάδιο Ασκήσεων

Άσκηση 1:

1.) Εφ' όσον η Eve γνωρίζει ότι χρησιμοποιούν την μέθοδο Vigenere, μπορεί αρχικά να εκτιμήσει το μήκος του κλειδιού εντοπίζοντας επαναλαμβανόμενες συμβολοσειρές στο κρυπτοκείμενο (που αντιστοιχούν σε ίδια γράμματα-λέξεις), και υπολογίζοντας τον ΜΚΔ της περιόδου εμφάνισης όλων των μοτίβων αυτών. Στην συνέχεια, γνωρίζοντας το μήκος κλειδιού μπορεί να βρει τα γράμματά του εφαρμόζοντας στατιστική ανάλυση στο κείμενο, με εμπειρικές γνώσεις για τα συχνότερα και σπανιότερα εμφανιζόμενα γράμματα στην αγγλική γλώσσα (ή οποιαδήποτε άλλη).
Η Eve θα μπορεί να ακολουθήσει την ίδια ακριβώς διαδικασία για ένα κρυπτογραφημένο κλειδί γνωστού μήκους, για κάθε μία όμως από τις 26 δυνατές μεταθέσεις του Caesar cipher. Επομένως, με το επιπλέον αυτό βήμα οι Alice και Bob αυξάνουν τον όγκο των απαραίτητων υπολογισμών, αλλά όχι την δυσκολία του προβλήματος από άποψη υπολογισιμότητας.

2.) Επειδή το αρχικό κλειδί είναι γνωστό, αρκεί να δοκιμαστούν όλες οι δυνατές μεταθέσεις της λέξης cryptography έως ότου βρεθεί αληθινό κείμενο. Για την άσκηση έγιναν δύο παραδοχές: πρώτον, όλα τα γράμματα μετατρέπονται σε κεφαλαία για ευκολία στον προγραμματισμό, δεδομένου ότι αυτό δεν επηρεάζει την ουσία του μηνύματος. Δεύτερον, επειδή στο δοθέν κρυπτοκείμενο τα κενά και τα σημεία στίξης δεν έχουν κρυπτογραφηθεί, αυτά αγνοήθηκαν.

```
ex1_text = "Nd Dhy. A dcmgv yk ccob xsieewa svptdwn os ptp Kqg, url gz
wazwry vaffu jj t mgzogk tsi os xyextrm lmb hildcmzu. B plsgp plpz oq
npw dci Otikigkb usklxc.Egi ahr lrdrd zh g rcr qg wvox zwx hglpsqzw
bxrunubydo os wpextrm cgb cik?"
```

cryptoex1.py

```
## encrypt key with Caesar cipher
def caesar_encrypt(text: str, shift: int) -> str:
    return ''.join([chr((ord(i) + shift - ord('A')) % 26 + ord('A'))
for i in text])

def caesar_decrypt(encrypted_text: str, shift: int) -> str:
    return ''.join([chr((ord(i) - shift + ord('Z') + 1) % 26 +
ord('A')) for i in encrypted_text])

def vig_encrypt(text: str, key: str) -> str:
    encrypted_text = ''
    for i in range(len(text)):
```

```
        encrypted_text += caesar_encrypt(text[i], ord(key[i %
len(key)]) - ord('A'))

    return encrypted_text

def vig_decrypt(encrypted_text: str, key: str) -> str:
    decrypted_text = ''
    offset = 0 ## pointer offset to account for punctuation and
whitespace
    for i in range(len(encrypted_text)):
        if(not encrypted_text[i].isalpha()):
            decrypted_text += encrypted_text[i]
            offset += 1
        else:
            decrypted_text += caesar_decrypt(encrypted_text[i],
ord(key[(i - offset) % len(key)]) - ord('A'))

    return decrypted_text

if(__name__ == "__main__"):
    ex1_key = "CRYPTOGRAPHY"
    enkey = caesar_encrypt(ex1_key, 4)
    decrypted_text = vig_decrypt(ex1_text.upper(), enkey)
    print("shift: {}".format(4))
    print(decrypted_text)
```

Η τιμή 4 για την μετάθεση προέκυψε δοκιμάζοντας όλες τις δυνατές μεταθέσεις 1 έως 25 (και την τετριμμένη 0). Εκτελώντας το παραπάνω λαμβάνουμε:

```
shift: 4
HI BOB. I THINK WE HAVE FINALLY MANAGED TO WIN EVE, BUT WE SHOULD
THINK OF A BETTER WAY TO ENCRYPT OUR MESSAGES. I STILL WANT TO USE THE
VIGENERE CIPHER.CAN YOU THINK OF A WAY TO MAKE OUR MESSAGES IMPOSSIBLE
TO DECRYPT FOR HER?
```

3.) Προκειμένου το κρυπτοσύστημα Vigenere να γίνει εντελώς απρόσβλητο σε αποκρυπτογράφηση, πρέπει να επιλεγεί κλειδί με μήκος ίσο με αυτό του κειμένου (one-time pad).

cryptoex1_2.py

```
from cryptoex1 import caesar_encrypt, vig_encrypt

answer = "HI ALICE. WE MUST USE A KEY OF LENGTH EQUAL TO THAT OF THE
MESSAGE, THAT IS, A ONE TIME PAD"
key = "CRYPTOGRAPHY"
```

```
enkey = caesar_encrypt(key, 14)

print(vig_encrypt(answer, enkey))
```

Εκτελώντας λαμβάνουμε

```
XNFDSKWJVWRQJRGVAVOXSWVFAJKWVHNQSQBFXYQTBCFYHROFXFFWVHNYVHOYUXEDNGZYHK
VFJNEIACNTBHOFYRQWWCX
```

Άσκηση 2:

α.) Η μέθοδος αυτή πράγματι αυξάνει την δυσκολία αποκρυπτογράφησης του κειμένου. Για την εύρεση του μήκους του κλειδιού μπορεί να χρησιμοποιηθεί η ίδια μέθοδος με αυτήν για την αποκρυπτογράφηση του Vigenere, με ίδια πολυπλοκότητα, αφού στην ουσία πρόκειται για ίδια κρυπτογράφηση με μεγαλύτερο κλειδί (οι διαδοχικές ολισθήσεις περιοδικά θα παράγουν την ίδια συμβολοσειρά – κλειδί). Όμως, στην συνέχεια αυξάνεται η υπολογιστική δυσκολία καθώς χρειάζεται να βρεθούν τα γράμματα κλειδιού (πολύ) μεγαλύτερου μήκους. Οι χειρότερες τιμές για τον k είναι οι τετριμμένες: 0, και τα ακέραια πολλαπλάσια του μήκους του κλειδιού, που ουσιαστικά δεν έχουν επίδραση στην ολίσθηση. Αν το μήκος του κλειδιού και το k είναι πρώτοι μεταξύ τους μεγιστοποιείται η «περίοδος» της προκύπτουσας συμβολοσειράς, μετά από πόσες ολισθήσεις, δηλαδή, επαναλαμβάνεται το μοτίβο.
β.) Όπως αναφέρθηκε, ο υπολογισμός του μήκους του κλειδιού είναι εύκολος υπολογιστικά. Στην συνέχεια, προκειμένου να μην χρειαστεί να γίνει αμέσως στατιστική κρυπτανάλυση για την εύρεση των γραμμάτων, όπως ακριβώς για το σύστημα Vigenere, η οποία θα είχε αποτέλεσμα μεν, χρονοβόρο στον υπολογισμό δε, μπορούμε να εξετάσουμε τις πιθανές τιμές του k, και να περιορίσουμε το πεδίο αναζήτησης χρησιμοποιώντας δείκτη σύμπτωσης. Υποθέτοντας ότι οι αντίπαλοι είναι έξυπνοι και χρησιμοποιούν βέλτιστη τιμή για το k, δηλαδή τέτοιο ώστε $\gcd(k, r) = 1$, όπου r το μήκος κλειδιού, περιορίζουμε περισσότερο τις πιθανές τιμές.

Άσκηση 3:

Αν $|\mathcal{L}| = t$ τότε $\mathbb{E}[I_r] \cong \sum_1^t \left(\frac{1}{t}\right)^2 = t \cdot \left(\frac{1}{t}\right)^2 = \frac{1}{t}$

Άσκηση 4:

Για την αποκρυπτογράφηση ακολουθείται η μέθοδος που περιεγράφη στην άσκηση 1. Η βασική ιδέα για την εύρεση του μήκους του κλειδιού είναι ότι, ίδιες λέξεις του αρχικού κειμένου σε απόσταση πολλαπλάσια του μήκος του κλειδιού r θα κωδικοποιούνται με ίδιον τρόπο, επομένως υπολογίζοντας των ΜΚΔ των αποστάσεων μεταξύ επαναλαμβανόμενων μοτίβων βρίσκουμε μια εκτίμηση για το μήκος του κλειδιού. Στην συνέχεια μπορούμε να χωρίσουμε το κείμενο σε r στήλες (όπου κάθε μία θα έχει κρυπτογραφηθεί με ίδια μετάθεση) και να υπολογίσουμε τον δείκτη σύμπτωσης κάθε μιας. Αν η τιμή του προσεγγίζει αυτήν του κειμένου φυσικής γλώσσας, αυτό θα σημαίνει ότι η στήλη αυτή μπορεί να έχει προκύψει από μετάθεση Caesar ενός κειμένου

φυσικής γλώσσας. Αφού επιβεβαιωθεί το μήκος του κλειδιού, ο κάθε χαρακτήρας αυτού μπορεί να βρεθεί με στατιστική κρυπτανάλυση βάσει της συχνότητας εμφάνισης των διαφόρων γραμμάτων.

Για το πρόγραμμα χρησιμοποιήθηκε ο προσεγγιστικός τύπος $r \cong \frac{I_{eng}-I_{rand}}{I_{text}-I_{rand}}$, και για την στατιστική κρυπτανάλυση η υποψήφια τιμή μετάθεσης απορρίφθηκε αν οποιοδήποτε από τα γράμματα E, A, I εμφανιζόταν σπανιότερα από οποιοδήποτε από τα X, Z, και Q.

cryptoex4.py

```
import sys
import math
from string import ascii_uppercase
from cryptoex1 import vig_decrypt, caesar_decrypt
import itertools

def IC(text):
    n = len(text)
    Itext = 0 ## index of coincidence
    for l in ascii_uppercase:
        fi = text.count(l)
        Itext += (fi * (fi - 1)) / (n * (n - 1))
    return Itext

def getColumns(text, r):
    res = []
    for column in range(r):
        column_text = ''.join([text[column + i * r] for i in
range(math.floor(len(text) / r))])
        res.append(column_text)
    return res

try:
    f = open(sys.argv[1], 'r')
    encrypted_text = f.read().replace('\n', '')
except:
    sys.exit()

Ieng = 0.065
Irand = 0.038
ic = IC(encrypted_text)
rest = (Ieng - Irand) / (ic - Irand) ## estimation for key length,
result 6.980961454529971

candidates_list = []
## Εξετάζουμε εύρος τιμών γύρω από την εκτίμηση Friedman
```

```
for r in range(math.ceil(rest) - 2, math.ceil(rest) + 3):
    for column_text in getColumns(encrypted_text, r):
        if(IC(column_text) < 0.05):
            print("key length %d rejected" % r)
            break

        ## για μια στήλη με αποδεκτό δείκτη συμπτώσεως, δοκιμάζουμε
όλες τις δυνατές μεταθέσεις και
        ## αποθηκεύουμε τις αποδεκτές σε string
        ## assume most frequent letters in the english language are
EAI, least frequent XZQ
        candidates = ''
        for k in ascii_uppercase:
            flag = True
            candidate_text = caesar_decrypt(column_text, ord(k) -
ord('A'))
            for (i, j) in itertools.product("EAI", "XZQ"):
                if(candidate_text.count(i) < candidate_text.count(j)):
                    flag = False
            if(flag): candidates += k

        candidates_list.append(candidates)

## όλοι οι δυνατοί συνδυασμοί των αποδεκτών γραμμάτων για κάθε στήλη
potential_keys = [''.join(k) for k in
itertools.product(*candidates_list)]

for pk in potential_keys:
    print("Key: ", pk)
    print(vig_decrypt(encrypted_text, pk))
    print()
```

Αποθηκεύουμε το κρυπτοκείμενο στο αρχείο ex4.txt και εκτελούμε `cryptoex4.py ex4.txt`:

```
key length 5 rejected
key length 6 rejected
key length 7 rejected
key length 8 rejected
Key:  KHALALGIB
ANDAEOMANWHOHMLDABABEAOAINSTHERJOSOMSAIDAPEAKTOUSWFCHILDREVANDHESAILYO
URCHILLRENARENOBYOURCHILLRENTHEYAZETHESONSINDDAUGHTMRSOFLIFEALONGINGFW
RITSELFTPEYCOMETHZOUGHYOUBCTNOTFROMGOUANDTHOCGHTHEYARMWITHYOUYMTTHEYBE
LWNGNOTTOYWUYOUMAYGQVETHEMYOCRLOVEBUTVOTYOURTHWUGHTSFORBHEYHAVETPEIROW
NTHWUGHTSYOUUAYHOUSETPEIRBODIEABUTNOTTHMIRSOULSFWRTHEIRSOCLSDWELLIVTHE
HOUSEWFTOMORROEWHICHYOUKANNOTVISQTNOTEVENQNYOURDREIMSYOUMAYATRIVETOBML
IKETHEMJUTSEEKNOBTOMAKETHMMLIKEYOUNORLIFEGOMSNOTBACKEARDNORTAZRIESWITH
```

GESTERDAYGOUARETHEJOWSFROMWPICHYOURCPILDRENASTIVINGARRWWSARESENBFORTHT
HEIRCHERSEEATHEMARKUXONTHEPATPOFTHEINFQNITEANDHMBENDSYOUEITHHISMIOHTTH
ATHIAARROWSMAGGOSWIFTAVDFARLETYWURBENDINOINTHEARCPERSHANDBMFORGLADNMSS
FOREVEVASHELOVEATHEARROWBHATFLIESAOHELOVESILSOTHEBOETHATISSTIBLE

Key: KHALALGIF
ANDAEOMAJWHOHMLDAXABEAOAINOTHERJOSOISAIDAPEAGTOUSWFCHELDREVANDDESAILYO
UNCHILLRENWRENOBYOUNCHILLRENPHEYAZETHASONSINDDWUGHTMRSOBLIFEALONCINGFW
RITOELFTPEYCKMETHZOUGDYOUBCTNOPFROMGOUAJDTHOCGHTDEYARMWITDYOUYMTTHAYBE
LWNGNKTTOYWUYOQMAYGQVETDEMYOCRLOREBUTVOTYKURTHWUGHPSFORBHEYDAVETPEIRKW
NTHWUGHPSYOUUAYHKUSETPEIRXODIEABUTJOTTHMIRSKULSFWRTHAIRSOCLSDSELLIVTHE
DOUSEWFTOIORROEWHIYHYOUKANNKTVISQTNOPEVENQNYOQRDREIMSYKUMAYATRIRETOBML
IKATHEMJUTSAEKNOBTOMWKETHMMLIGEYOUNORLEFEGOMSNOPBACKEARDJORTAZRIEOWITH
GESTARDAYGOUANETHEJOWSBROMWPICHUOURCPILDNENASTIVIJGARRWWSANESENBFORPHT
HEIRCHARSEEATHEIARKUXONTDEPATPOFTDEINFQNITAANDHMBENZSYOUEITHDISMIOHTTD
ATHIAARRKWSMAGGOSSIFTAVDFANLETYWURBANDINOINTDEARCPERSDANDBMFORCLADNMSS
FKREVEVASHALOVEATHEWRROWBHATBLIESAOHEHOVESILSOPHEBOETHAPISSTIBLE

Key: KHALILGIB
ANDAWOMANWHOHELDABABEAGAINSTHERBOSOMSAIDSPEAKTOUSOFCHILDRENANDHESAIDYO
URCHILDRENARENOTYOURCHILDRENTHEYARETHESONSANDDAUGHTERSOFLIFESLONGINGFO
RITSELFTHEYCOMETHROUGHYOUBUTNOTFROMYOUANDTHOUGHTHEYAREWITHYOUYETTHEYBE
LONGNOTTOYOUYOUMAYGIVETHEMYOURLOVEBUTNOTYOURTHOUGHTSFORTHEYHAVETHEIROW
NTHOUGHTSYOUMAYHOUSETHEIRBODIESBUTNOTTHEIRSOULSFORTHEIRSOULSDWELLINTHE
HOUSEOFTOMORROWWHICHYOUCANNOTVISITNOTEVENINYOURDREAMSYOUMAYSTRIVETOBEL
IKETHEMBUTSEEKNOTTOMAKETHEMLIKEYOUFORLIFEGOESNOTBACKWARDNORTARRIESWITH
YESTERDAYYOUARETHEBOWSFROMWHICHYOURCHILDRENASLIVINGARROWSARESENTFORTHT
HEARCHERSEESTHEMARKUPONTHEPATHOFTHEINFINITEANDHEBENDSYOUWITHHISMIGHTTH
ATHISARROWSMAYGOSWIFTANDFARLETYOURBENDINGINTHEARCHERSHANDBEFORGLADNESS
FOREVENASHELOVESTHEARROWTHATFLIESSOHELOVESALSOTHEBOWTHATISSTABLE

Key: KHALILGIF
ANDAWOMAJWHOHELDAXABEAGAINOTHERBOSOISAIDSPEAGTOUSOFCHELDRENANDDESAIDYO
UNCHILDRENWRENOTYOUNCHILDRENPHEYARETHASONSANDDWUGHTERSOBLIFESLONCINGFO
RITOELFTHEYCKMETHROUGDYOUBUTNOPFROMYOUAJDTHOUGHTDEYAREWITDYOUYETTHAYBE
LONGNKTTOYOUYOQMAYGIVETDEMYOURLOREBUTNOTYKURTHOUGHPSFORTHEYDAVETHEIRKW
NTHOUGHPSYOUMAYHKUSETHEIRXODIESBUTJOTTHEIRSKULSFORTHAIRSOULSDSELLINTHE
DOUSEOFTOIORROWWHIYHYOUCANNKTVISITNOPEVENINYOQRDREAMSYKUMAYSTRIRETOBEL
IKATHEMBUTSAEKNOTTOMWKETHEMLIGEYOUFORLEFEGOESNOPBACKWARDJORTARRIEOWITH
YESTARDAYYOUANETHEBOWSBROMWHICHUOURCHILDNENASLIVIJGARROWSANESENTFORPHT
HEARCHARSEESTHEIARKUPONTDEPATHOFTDEINFINITAANDHEBENZSYOUWITHDISMIGHTTD
ATHISARRKWSMAYGOSSIFTANDFANLETYOURBANDINGINTDEARCHERSDANDBEFORCLADNESS
FKREVENASHALOVESTHEWRROWTHATBLIESSOHEHOVESALSOPHEBOWTHAPISSTABLE

Με το κλειδί KHALILGIB (το όνομα του συγγραφέως) παράγεται το πραγματικό αρχικό κείμενο-μήνυμα, το ποίημα “On Children”.

Άσκηση 5:

1.) Το κρυπτοσύστημα της σελίδας 21 περιγράφεται ως εξής: $\mathcal{M} = \{0,1\}, \mathcal{C} = \{A,B\}, \mathcal{K} = \{K_1, K_2\}, \mathbb{P}[K_1] = \frac{1}{3}, \mathbb{P}[K_2] = \frac{2}{3}$. Σύμφωνα με τον ορισμό του Shannon, ένα κρυπτοσύστημα χαρακτηρίζεται από τέλεια μυστικότητα αν:
$\forall x \in \mathcal{M}, y \in \mathcal{C}: \mathbb{P}[M = x | C = y] = \mathbb{P}[M = x]$, όπου M και C είναι τυχαίες μεταβλητές που λαμβάνουν τιμές στα $\mathcal{M}, \mathcal{C}$ αντιστοίχως.
Θεωρώντας τυχαία μεταβλητή $K \in \mathcal{K}$, και χρησιμοποιώντας τον κανόνα Bayes, για το δοθέν κρυπτοσύστημα ισχύει:

$$\mathbb{P}[M = 0 | C = B] = \frac{\mathbb{P}[C = B | M = 0]\mathbb{P}[M = 0]}{\mathbb{P}[C = B]} =$$

$$= \frac{\mathbb{P}[K = K_2]\mathbb{P}[M = 0]}{\mathbb{P}[K = K_1]\mathbb{P}[M = 1] + \mathbb{P}[K = K_2]\mathbb{P}[M = 0]} = \frac{\mathbb{P}[M = 0]}{\frac{\mathbb{P}[M = 1]}{2} + \mathbb{P}[M = 0]} > \mathbb{P}[M = 0]$$

διότι $\frac{\mathbb{P}[M=1]}{2} + \mathbb{P}[M = 0] < 1$ εφ' όσον $\mathbb{P}[M = 1] + \mathbb{P}[M = 0] = 1$
Άρα δεν διαθέτει τέλεια μυστικότητα.

2.) Θα χρησιμοποιηθεί η ισοδύναμη συνθήκη τέλειας μυστικότητας:
$\forall x_1, x_2 \in \mathcal{M}, y \in \mathcal{C}: \mathbb{P}[C = y | M = x_1] = \mathbb{P}[C = y | M = x_2]$
Έστω κρυπτοσύστημα με $|\mathcal{M}| = |\mathcal{C}| = |\mathcal{K}|$. Για να έχει τέλεια μυστικότητα είναι αναγκαίες οι εξής συνθήκες:
α) για κάθε $x \in \mathcal{M}, y \in \mathcal{C}$ υπάρχει μοναδικό $k \in \mathcal{K}$, ώστε $enc_k(x) = y$
β) κάθε κλειδί επιλέγεται με την ίδια πιθανότητα $1/|\mathcal{K}|$
Απόδειξη: παραβίαση της α' οδηγεί σε μηδενική δεσμευμένη πιθανότητα κάποιου y, δεδομένου x, δηλαδή άτοπο. Από αρχή περιστεριώνα και ιδιότητα 1-1 της enc προκύπτει $\forall y \in \mathcal{C}, k_1, k_2 \in \mathcal{K}, \exists x_1, x_2 \in \mathcal{M}: enc_{k_1}(x_1) = enc_{k_2}(x_2) = y$, και χρησιμοποιώντας την ισοδύναμη συνθήκη παραπάνω καταλήγουμε στο συμπέρασμα ότι τα k_1 και k_2 είναι ισοπίθανα.
Επομένως, αν οι χώροι είναι ισοπληθικοί, κάθε κλειδί πρέπει να επιλέγεται με ίση πιθανότητα. Αν, όμως, δεν είναι ισοπληθικοί, και ισχύει $|\mathcal{M}| < |\mathcal{C}| < |\mathcal{K}|$, τότε ένα υποσύνολο κλειδιών $\mathcal{K}'$ με $|\mathcal{K}'| = |\mathcal{K}| - |\mathcal{C}|$ μπορεί να μην χρησιμοποιείται καθόλου, και η ιδιότητα της τέλειας μυστικότητας θα διατηρείται.

3.) i. $\forall x \in \mathcal{M}, y \in \mathcal{C}: \mathbb{P}[C = y] = \mathbb{P}[C = y | M = x]$
Χρησιμοποιώντας τον κανόνα Bayes και την παραπάνω σχέση προκύπτει:

$$\mathbb{P}[C = y | M = x] = \frac{\mathbb{P}[M = x | C = y]\mathbb{P}[C = y]}{\mathbb{P}[M = x]} \Rightarrow \mathbb{P}[C = y] = \frac{\mathbb{P}[M = x | C = y]\mathbb{P}[C = y]}{\mathbb{P}[M = x]} \Rightarrow$$

$$\Rightarrow \frac{\mathbb{P}[M = x | C = y]}{\mathbb{P}[M = x]} = 1 \Rightarrow \mathbb{P}[M = x | C = y] = \mathbb{P}[M = x]$$

που είναι ο ορισμός κατά Shannon. Η απόδειξη της αντίστροφης συνεπαγωγής είναι προφανής.

ii. $\forall x_1, x_2 \in \mathcal{M}, y \in \mathcal{C}: \mathbb{P}[C = y | M = x_1] = \mathbb{P}[C = y | M = x_2]$
Αρκεί να αποδειχθεί ότι η παραπάνω ισοδυναμεί με την σχέση της i.
Ευθεία συνεπαγωγή:
$\mathbb{P}[C = y] = \mathbb{P}[C = y | M = x] \Rightarrow \forall x_1, x_2 \in \mathcal{M}: \mathbb{P}[C = y | M = x_1] = \mathbb{P}[C = y]$ και
$\mathbb{P}[C = y | M = x_2] = \mathbb{P}[C = y]$ άρα $\mathbb{P}[C = y | M = x_1] = \mathbb{P}[C = y | M = x_2]$

Αντίστροφη συνεπαγωγή:
$\forall x_1, x_2 \in \mathcal{M}, y \in \mathcal{C}: \mathbb{P}[C = y|M = x_1] = \mathbb{P}[C = y|M = x_2] \Rightarrow \mathbb{P}[C = y|M = x] = \lambda, \forall\, x \in \mathcal{M}$
όπου λ σταθερά.
Άρα, εφ' όσον $\mathbb{P}[C = y] = \sum_{x \in \mathcal{M}} \mathbb{P}[C = y|M = x]\mathbb{P}[M = x]$, ισχύει

$$\mathbb{P}[C = y] = \sum_{x \in \mathcal{M}} \lambda \mathbb{P}[M = x] = \lambda \sum_{x \in \mathcal{M}} \mathbb{P}[M = x] = \lambda \cdot 1 = \mathbb{P}[C = y|M = x]$$