```
דחיסת דקדוק Re-Pair
                                                                       הגדרות לאלגוריתמים של עץ skeleton:
  כל עוד ישנו צמד תווים שחוזר על עצמו
                                      קוראים את הטקסט תו אחרי תו ומוסיפים לS.
                                                                                                                                                                                                                         פר, דר נכתב ע"י צבי מינץ ויואב ג
בלא דר נכתב ע"י צבי מינץ ויואב ג
בלא דרצה :פרופ' דנה שפירא
                                  1. אם צמד תווים מופיע פעמיים בצד ימיו הוא יוחלף
                                                                                                                                                                                                                    נכתב ע"י צבי מינץ ויואב גרוס
                                                                        base(i) = 2(base(i-1) + n_{i-1})
                                                                                                              Golomb decode(b)
                                                                                                                                                Golomb encode(x,b)
 מצא את הזוג השכיח ביותר והחלף אותו
                                                              בכלל חדש.
                               2. אם משתנה מופיע פעם אחת בצד ימין – נמחק אותו.
                                                                              seq(i) = seq(i-1) + n_{i-1} \quad q \leftarrow Unary\_decode()-1;
                                                                                                                                                      a \leftarrow (x-1) \text{ div } b:
                  (בהינתן מילון עם מילות קוד באורך קבוע, יש להתאים מחרוזות) Tunstall
                                                                             diff(i) = base(i) - seq(i) r\leftarrowMinimal binary_decode(b); r\leftarrowx-q·b;
                                                                                                                                                                                        (בסוף רוצים קוד K(C) \le 1 מינמלי E(C,P)ו-
                                                                                                                                                                                         E(C,P) = \sum_{i=1}^{n} p_i \cdot |c_i| אורך מילת הקוד הממוצעת:
                                                                                                                                                     I = Unary\ encode(q+1);
                                                                                                              return r+q\cdot b;
                                                                         :Decoding Algorithm
                                             כל עוד נותר מקום לΣ תווים נוספים:
                                                                                                                                                                                                                    I(s_i) = -\log_2 p_i אינפורמציה:
                                                                                                                                                     n = Minimal_binary_encode(r,b);    return l·n
                                                                        1. tree_pointer<-root
         . למילון, הכנס את \{d\sigma\colon\sigma\in\Sigma\} מילת הקוד עם ההסתברות המקסי' במילון, הכנס את 1
                                                                                                                                                                                         H(P) = -\sum_{i=1}^{n} p_i \log_2 p_i אנטרופיה (חסם תחתון):
                                                                                                                             נבחר את b להיות Rice code: נבחר את
                                                                                                        Skeleton
                                             .2 אם Z \notin \Sigma מחק את d מהמילון.
                                                                                                                                                                                                                         \forall C: H(P) \leq E(C,P) כלומר
                                                                                                                   חלק ראשון: קוד אונרי של (right shift k bits)) של 1+(x-1
                                                           קוד פיבונאצ'י 1
                                                                                                                                                                                         C' אזי קיים קוד K(C) = \sum_{i=1}^{n} 2^{-|c_i|} \le 1 אזי קיים קוד
                                                                        4. while i<length of string
                                                                                                                       X-1 (לא תמיד הבי יעיל) Shannon-Fano Algorithm:
                   כתוב את המספר בבסיס פיבונאצ'י, הוסף 1 משמאל, הפוך את המספר.
                                                                                                                                                                                          יסר ש-C'-ו ו|C| = |C'|, E(C, P) = E(C', P) פרפיקסי
                                                                                 if string[i]=0
 k+1 הן באורך אורך הקוד בטווח קוד מילות הקוד האורך הן באורך הורך הרF_{k+1} \leq j < F_{k+2} הון מילות מיידי. כל
                                                                                                                                                 מיין את ההסתברויות בסדר יורד.
                                                                                      tree_pointer<-left(tree_pointer)
                                                                                                                                                                                           אזי לא היים הוד פרפיקסי עם האורכים K(C) > 1
                                                           2 קוד פיבונאצ'י 4.2.
                                                                                 else tree pointer<-right(tree poi
                                                                                                                                  כל עוד קבוצת ההסתברות מכילה יותר מתו 1:
                                                                                                                                                                                                     קוד שלם: K(C) = 1. קוד שלם יוצר עץ מלא
                                             Fib2(n+1) עבור, Fib2(1) = 1 4.3.
                                                                                 if value(tree pointer)>0
                                                                                                                                                                                         (ניתן לפענוח בצורה יחידה) UD קוד חסר רישות \Rightarrow קוד
                                                                                                                    -נחלק את הקבוצה ל2 חלקים כך שסכום ההסתברויות בכל
                                                                                      codeword<-string [start ...
                      . משמאל מחוב את n בבסיס פיבונאצ'י, הפור את המספר, הוסף 10 משמאל.
                                                                                                                                                                                     קוד מיידי ⇔ קוד חסר רישות, קוד UD ≠קוד חסר רישות
                                                                                                                     חלק פחות או יותר זהה. -קבוצה אחת תקבל 1, והשנייה 0.
                                                      (start+value(tree_pointer)-1)]
                                                                                                                                             קוד הפמן: אלגוריתם לבניית עץ קנוני
                                                                                                                                                                                                                     מבחו לזיהוי קוד יחודי: (אם"ם)
       את לחפש אז לחפש את \sigma נמצא אז לחפש את – אפשר ע"י גרידי (אם 4.3.2.
                                                                                      output<-table[I(codeword)-
                                                                                                                        (l_1, ..., l_n) אחרי הרצת הפמן קיבלנו את כל האורכים
              וכו׳ ואז להתחיל מחדש),אפשר למצוא אופטימלי \sigma \cdot \sigma_2
                                                                        diff[value(tree_pointer)]]
                                                                                                                                                                                  • Examine all pairs of codewords:
                                                                                                                                                                                                                                                                 היא רישא של x ולכן סיפא מתנדנדת היא \sigma אם x היא מילת קוד מקורית – הקוד לא ייחודי
                                                                                                                        1.maxlength ←האורך של המילה הארוכה ביותר
 4.3.3. ע"י רדוקציה לגרף. דחיסה אדפטיבית – המילון נבנה בזמן הקידוד
                                                                                      tree_pointer←root
                                                                                                                                                                                  1. Construct a list of all codewords.
    ומכיל שלשות Sliding Window, Look Ahead Buffer עם – LZ77
                                                                                      start<-start+value(tree pointer) 2. for i = 1 to maxlength do \{num[i] = 0\}
                                                                                                                                                                                  2. If there exist a codeword, a, which is a prefix of
                  (0,0,\sigma) כאשר לקדד תו בודד זה (off,len,symbol)
                                                                        4.3.5
                                                                                      i<-start
                                                                                                                        3. for i = 1 to n do \{num[l_i] + +\}
                                                                                                                                                                                      another codeword, b, add the dangling suffix to the
                                        : LZ77 אלגוריתם קידוד
                                                                        4.4. else i++
1. tree_pointer<-root
                                                                                                                                                                                      list (if it is not there already), until:
                                                                                                                        4.firstcode[maxlength] \leftarrow 0
         1. p←1 // The next character to be coded
                                                                                                                                                                                         You get a dangling suffix that is an original codeword \rightarrow the
                                                                                                                        5. for i = maxlength - 1 downto 1 do
                                                                        2. i <- start<-1
         2. while there is text remaining to be coded
                                                                        3. while i<length_of_string
                                                                                                                        5.1.firstcode[i] \leftarrow (firstcode[i+1] +
                                                                                                                                                                                         There are no more unique dangling suffixes \rightarrow the code is
         2.1. search for the longest match for S[p..] in S[p-
                                                                               if string[i]=0 tree pointer<-
                                                                                                                                                              num[i+1])/2
         W...p-1] if the match occurs at position m with
                                                                        left(tree pointer)
                                                                                                                        6. for i = 1 to maxlength do
                                                                                                                                                                                            מודלים: (ללמוד ולעשות הנחות על מבנה הטקסט)
         length I
                                                                               else tree pointer<-right(tree pointer)
                                                                                                                                                                                                 Lossy - Lossless – יש 2 סוגים של דחיסות
ex.Video ex.Text
                                                                                                                        6.1. nextcode[i] \leftarrow firstcode[i]
                                                                               if value(tree_pointer)>0
         2.2 Output the triple (p-m, I,S[p+l])
                                                                                                                        7. for i = 1 to n do
                                                                        3.3.1.
                                                                                  len<-value(tree_pointer)
                           length > offset הצבה עצמית – כאשר
                                                                        3.3.2.
                                                                                  codeword<-string[start...(start+len-1)]
                                                                                                                                                                                (prelude אין |\Sigma| = 256(Ascii) כאשר p_i = \frac{1}{|\Sigma|} (אין
                           (AVL) שיפור, משתמש בעץ בינארי מאוזן – LZSS
                                                                                                                        7.1.codeword[i] \leftarrow nextcode[l_i]
                                                                        3.3.3.
                                                                                  if flag(tree_pointer)=1 and
                           ומחזיק ביט + (off,len) או ביט + אסקי (תו בודד)
                                                                                                                        7.2.symbol[l_i, nextcode[l_i] - firstcode[l_i]] \leftarrow i
 w ← first char of input
                                                                                                                                                                                    |\Sigma|= כך p_i=rac{1}{|\Sigma|} בטקסט שונים בטקסט p_i=rac{1}{|\Sigma|}
                           Index | Phrase מחזיק, trie שימוש בעץ – LZW
                                                                        2I(codeword)>=base(len+1)
                                                                                                                        7.3.nextcode[l_i] + +
 repeat
                      ומאותחל עם תווים בודדים, בנוסף מחזיק טבלה w | k
                                                                        3.3.3.1
                                                                                      codeword<-string[start...(start+len)
                                                                                                                                                                                E(C,P)=H(P)+rac{8\cdot|\Sigma|+8}{|Text\,Size|}ולכן ולכן אחבסס על מודל אסקיE(C,P)=H(P)+rac{8\cdot|\Sigma|+8}{|Text\,Size|}
                                                     אלגוריתם לקידוד:
                                                                        3.3.3.2.
 k ← next char
                                                                                                                                 אלגוריתם לפענוח של הפמן קנוני ע"י הטבלאות:
                                                                        3.3.4.
   if (EOF) output code(w)
                                                                                 output<-table[I(codeword)-diff[len]]
                                                                                                                                                                                 סמי סטטי עם הסתברויות עצמיות: p_i = rac{
u_i}{m} כאשר מספר
                                                                                                                        1. v \leftarrow nextInputBit()// קבלת הביט הראשון
                                                                                 tree pointer<-root
                                                                                 i<-start<-start+len
                                                                                                                        2.i = 1
                                                                                                                                                                               ההופעות של s_i בטקסט בגודל m. נציין ב-p כמה עולה לעביר
     if ((w \cdot k) \in Dictionary) then w \leftarrow w \cdot k
                                                                        3.4. else i++
                                                                                                                        3. while v < firstcode[i] do
                                                                                                                                                                                               E(C,P) = H(P) + \frac{8\cdot|\Sigma|\cdot|P|+8}{|Text\ Size|}הסתברות ולכן
              output code(w)
              Dictionary \leftarrow w \cdot k
                                                                                               ow \_ bound (s_i) \leftarrow \sum_{i=1}^{n} p_i
                                                                                                                        3.1. v \leftarrow 2v + nextInputBit()
                                                                         [0,1) קידוד - אַריתמטי
                                                                                                                                                                                                    קודים: (רוצים למצוא קודים בסדר גודל לוגרתמי)
                                                                                                                        3.2. i + +
                                                                                              igh bound(s_i) \leftarrow \hat{\sum}
                                                                                                                        אם יצאנו מהלולאה אזי v מכילה מילת קוד תקינה//
                                                                                                                                                                                                       קודים מסדר ראשון: ישנה תלות בין כל זוג תווים
                                                                        high \leftarrow 1.0
                                                                        while input symbols remain{
                                                                                                                        4.return\ symbol[i, v-firstcode[i]]
                                                                                                                                                                                                                   Unarv(X) = 1^{|X-1|} \cdot 0: Unary Code
אלגוריתם פענוח ל LZW: אתחל מילון עם תוים בודדים
                                                                           range ← high - low
                                                                                                                                        אלגוריתם לעדכון עץ הפמן דינמי:
OLD = first input code
                                                                                                                                                                                              [\log_2 n] כל סימן מייצג מילת קוד באורך: Binary Code
output translation of OLD
                                                                           high ← low + high_bound(symbol)*range
                                                                                                                                                                                       יהיו 2^{\lceil \log_2 n \rceil} - n אם יש: Minimal Binary Code
while not end of input stream{
                                                                           low ← low + low bound(symbol)*range
                                                                                                                           replace q by a parent 0-node with two 0-node
                                                                                                                                                                                        \lfloor \log_2 n \rfloor באורך באורך אורך ביטים ושאר באורך באורך באורך באורך באורך ו
 NEW = next input code
                                                                                                                           q = left child:
                                                                        Output any value in [low, high)
 if NEW is not in the string table
                                                                                                                       if q is a sibling of a 0-node
                                                                                                                                                                                     של מספר הביטים Unary גמה): החלק הראשון זה ביטים (גמה): C_{
m v}: Elias
     S = translation of OLD
                                                                                                                            interchange q with the highest numbered leaf
                                                                                 אלגוריתם לפענוח קוד אריתמטי בטווח [0.1]:
                                                                                                                         the same weight:
                                                                                                                                                                                          ללא '1' הראשון X ליצוג X והחלק השני זה קוד הבינארי של
     S = S \cdot C
                                                                       encoded ← Get (encoded number)
                                                                                                                           increment q's weight by 1;
                                                                                                                                                                                       X של מספר הביטים ליצוג \mathcal{C}_{\nu} של החלק הראשון זה \mathcal{C}_{\delta}
                                                                                                                           q = parent of q
                                                                        do{ Find symbol whose range contains encoded
                                                                                                                       while q is not root
     S = translation of NEW
                                                                       Output the symbol
                                                                                                                           interchange q with the highest numbered node
                                                                                                                                                                                                    והחלק השני זה קוד הבינארי של X ללא '1' הראשוו
  output S
                                                                                                                       of the same weight;
                                                                        range ← high(symbol) – low(symbol)
  C = first character of S
                                                                                                                           increment a's weight by 1:
                                                                                                                                                                                                                                                           סדרי גודל:
                                                                        encoded \leftarrow (encoded - low(symbol))/range
  Translation(OLD) C to the string table
                                                                                                                           q = parent of q
                                                                                                                                                                                       C_{v}: 1 + 2 \cdot \left[ \log_2 x \right] bits \ \[ C_{\delta}: 1 + 2 \left[ \log_2 \log_2 2x \right] + \left[ \log_2 x \right] bits
  OLD = NEW }
                                                                                                                       increment q's weight by 1
```