

Grupo Miércoles 10:00-12:00 semanas A

-Práctica 1-

Autor: Hector Toral

NIP: 798095

Ejercicio 1.1:

1. Resumen

He creado una secuencia de arranque que cuando detecta el carácter ">" ejecuta la secuencia de arranque "cabecera" comenzando la detección de la siguiente expresión regular ".\n" de tal manera que cogiese la cadena desde ">" hasta el salto de línea y así insertarlo directamente en el fichero nuevo.

Cuando la secuencia de arranque no detecta el carácter ">" procede a ejecutar el resto de expresiones regulares, la expresión que he usado ha sido "X+" siendo X uno de los 4 aminoácidos a comprimir en el fichero, cuando se detecta uno de estos, si hay más de 1 lo que hace es contarlos y escribir "X" seguido del número de aminoácidos iguales detectados y si es igual a 1 pues se deja ese aminoácido.

2. Pruebas

Para comprobar el ej 1.1 he realizado la compresión de los ficheros aportados en los materiales de la práctica

```
C4GCA2ACGT3G2TG2AC3TCAGAT2CA2CTG2CAGTA2C2AGA2TG2AGA2CGCAGTG4CGC
GATCA4CA2CGTCG2C4A2G2T3AC3A2TA2TACTGCGTCT2G2T2CAC2GCTCTCACTCA2CA
TG2CA2G2A2GAC2T2A3T2C3TCGAG2ACA2G2CGT2C2A2T2A2CAC2A2TAGCAGTC2AGATGAC
CA3T2G2CTACTAC2GA2GAGCTAC2AGACGA2T2CGTG2TG2TGACG2TA4TGA3GATCTCAGTC
CA2GATG2TAT3CTACTAC2TAG2A2CTG3C2AGA2GCTG2ACT2C3TATG2TGCTA2CA3GACG2
CATCATATG3T2GCA2CTGAG3AGC2T2GA2TACAC2A4GATCACAT2G2CAC3GCA2TC2TGCT
A2CA2TGCTGCA2TCGTGCTACA2CT2C2TCA2G2A2CA2CAT2GC2A4G2CT2CTACGCAGA2G3A
GCAGAG2CG2CAGTCA2GC2TCT2CTCGT2C2TCATCACGTAGTCGCA2CAGT2CA2GA3T2CA2CTC2
AG2CAGCAGTAG4A2CT2CTC2TGCTAGA2TG2CTG2CA2TG2CG2TGATGCTGCTCT2GCT3GCTG
CTGCT2GACAGAT2GA2C2AGCT2GAGAGCA4TGTCTG2TA3G2C2A2CA2CA2CA2G2C2A3CTG
TCACTA2GA3TCTGCTGCTGAG2CT2CTA2GA2GC2TCG2CA5CGTACTGC2ACTA3GCATACA2
TGTA2CACA2GCT3CG2CAGACGTG2TC2AGA2CA3C3A2G2A3T4G4AC2AG2A2CTA2TC
AGACA2G2A2CTGAT2ACA3CAT2G2C2GCA3T2GCACA2T3GC5AGCGCT2CAGCGT2CT2CG
GA2TGTGCGCAT2G2CATG2A2GTCACAC2T2CG3A2CGTG2T2GAC2TACACAG2TG2ATCA3T2
G2ATGACA3GATC2A3T3CA3GATCA2GTCAT4GCTGA2TA2GCATAT2GACGCATACA4CA
T2C3AC2A2CAGAGC2TA5G2ACA5GA2GA2G2CTGATGA3CTCA2GC2T2AC2GCAGAGAC
AGA2GA3CAGCA3CTGTGACTCT2CT2CTGCTGCAGAT3G2ATGAT3CTC2A3CA2T2GCA2CA
ATC2ATGAGCAGTGCTGACTCA2CTCAG2C2TA3CTCATGCAGAC2ACACA2G2CAGATG3CTATATA
A2CGT4CGCT4C2GT3ACGATATATAGTCTACTCT2GTGCAGA2TGA2T2CTCGTA2CTACATAG
CACA2GTAGATGTAGT2A2CT3A2TCTCACATAGCA2TCT3A2TCAGTGTGTA2CAT2AG3AG2ACT
TGA3GAGC2AC2ACAT4CAC2GAG2C2ACGCG2AGTACGATCGAGTGACAGTGA2CA2TGCTAG3
AGAGCTGC2TATATG2A2GAGC3TA2TGTGTA4T2A2T4AGTAGTGCTATC4ATGTGAT4A
ATAGCT2CT2AG2AGA2TGACGA3
```

Ejercicio 1.2:

1. Resumen

He creado una secuencia de arranque que cuando detecta el carácter ">" ejecuta la secuencia de arranque "cabecera" comenzando la detección de la siguiente expresión regular ".\n" de tal manera que cogiese la cadena desde ">" hasta el salto de línea y así insertarlo directamente en el fichero nuevo.

Cuando la secuencia de arranque no detecta el carácter ">" procede a ejecutar el resto de expresiones regulares, la primera expresión que he usado ha sido "[ACTG] [0-9]+" de tal manera que al detectar un aminoácido seguido de un número realice la descompresión. La segunda expresión es la misma pero sin el dato numérico de tal manera que así solo cojo el caso de que sea solo un aminoácido.

2. Pruebas

Para comprobar el ej 1.1 he realizado la descompresión del ficheros generado con el

```
CCCCGCATTACGTTTGGTGGACCCTCAGATTCAACTGGCAGTAACCAGAATGGAGAACGCAGTGGGGCGC
GATCAAAACAACGTCGGCCCCAAGGTTTACCCAATAATACTGCGTCTTGGTTCACCGCTCTCACTCAACA
TGGCAAGGAAGACCTTAAATTCCTCGAGGACAAGGCGTTCCAATTAACACCAATAGCAGTCCAGATGAC
CAAATTGGCTACTACCGAAGAGCTACCAGACGAATTCGTGGTGGTGACGGTAAAATGAAAGATCTCAGTC
CAAGATGGTATTTCTACTACCTAGGAAGCTGGGCCAGAAGCTGGACTTCCCTATGGTGCTAACAAAGACGG
CATCATATGGGTTGCAACTGAGGGAGCCTTGAATACACCAAAAGATCACATTGGCACCCGCAATCCTGCT
AACAATGCTGCAATCGTGCTACAACCTTCTCAAGGAACAACATTGCCAAAAGGCTTCTACGCAGAAAGGGA
GCAGAGGCGGCAGTCAAGCCTCTTCTCGTTCCTCATCAGTAGTCGCAACAGTTCAAGAAATTCAACTCC
AGGCAGCAGTAGGGGAACCTTCTCCTGCTAGAATGGCTGGCAATGGCGGTGATGCTGCTCTTGCTTTGCTG
CTGCTTGACAGATTGAACCAGCTTGAGAGCAAAATGTCTGGTAAAGGCCAACAACAAGGCCAAACTG
TCACTAAGAAATCTGCTGCTGAGGCTTCTAAGAAGCCTCGGCAAAAACGTAAGTCCACTAAAGCATACAA
TGTAACACAAGCTTTCGGCAGACGTGGTCCAGAACAAACCCAAGGAAATTTTGGGGACCAGGAATAATC
AGACAAGGAAGTATTACAAACATTGGCCGCAATTGCACAATTTGCCCCAGCGCTTCAGCGTTCTTCG
GAATGTCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTGGTTGACCTACACAGGTGCCATCAAATT
GGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGCTGAATAAGCATATTGACGCATACAAAACA
TTCCCAACCAACAGAGCCTAAAAAGGACAAAAAGAAGAAGGCTGATGAACTCAAGCCTTACCGCAGAGAC
AGAAGAAACAGCAAAGTGTGACTCTTCTTCTGCTGCAGATTTGGATGATTTCTCCAAACAATTGCAACA
ATCCATGAGCAGTGCTGACTCAACTCAGGCCTAACTCATGCAGACCACACAAGGCAGATGGGCTATATA
AACGTTTTTCGCTTTTCCGTTTACGATATATAGTCTACTCTTGTGCAGAATGAATTCTCGTAACTACATAG
CACAAGTAGATGTAGTTAACTTTAATCTCACATAGCAATCTTTAATCAGTGTGTAACATTAGGGAGGACT
TGAAAGAGCCACCACATTTTACCGAGGCCACGCGGAGTACGATCGAGTGTACAGTGAACAATGCTAGGG
AGAGCTGCCTATATGGAAGAGCCCTAATGTGTAAAATTAATTTTAGTAGTGCTATCCCCATGTGATTTTA
ATAGCTTCTTAGGAGAATGACGAAA
```

Ejercicio 2:

1. Resumen

2. Pruebas

Ejercicio 3:

1. Resumen

He modificado el AF que se nos daba en el guion de la práctica de tal manera que aceptase una diferencia ≤ 2 . La modificación consiste en añadir una fila más con la secuencia casa con las mismas transiciones que en el caso de ≤ 1 , he realizado el autómata en JFLAP, he unido los últimos 3 estados finales y he generado la expresión regular que acepta palabras con una distancia de Levenshtein ≤ 2 y posteriormente la he usado con flex.

2. Pruebas

Para la comprobación del ej3 he mirado por encima que no hubiese cosas raras como palabras de 16 letras ya que eso supera una distancia superior a 2. (No he realizado una prueba más exhaustiva debido a la falta de tiempo.)

ucase
vaca
vaga
vara
vasa
vasar
vaso
vaya
visa
yaba
yaca
yana
yapa
yaya
zaca
zafa
zaga
zala
zana
zapa
zas
zata
zaya

