

Tratamiento de Secuencias:

Estructuras de Iteración:

- 1- Repetir
- 2- Mientras
- 3- Iterar

Prof. Hilda Contreras

Programación 1

hildac.programacion1@gmail.com

Tratamiento de Secuencias

Contenido:

- Máquina de Adquisición de Secuencias de Caracteres (MAC)
- Estructuras de repetición o iteración:
 - Repetir
 - Mientras
 - Iterar
- Máquina de Creación de Secuencia de Caracteres (MCS)

Tratamiento de Secuencias

La información de los problemas reales es compleja viene dada generalmente en colecciones de elementos relacionados entre si. Por lo tanto, es necesario conocer diferentes técnicas para la estructuración de dicha información. La más sencilla y natural es la **Secuencia** (0 mas elementos del mismo tipo sobre los que esta definida una relación de orden).

Tratamiento de Secuencias

Necesitamos: tener acceso, realizar tratamiento y crear secuencias

Utilizamos un *Máquina* hipotética o abstracta que nos permite describir algoritmos para tratar secuencias haciendo abstracción de su representación

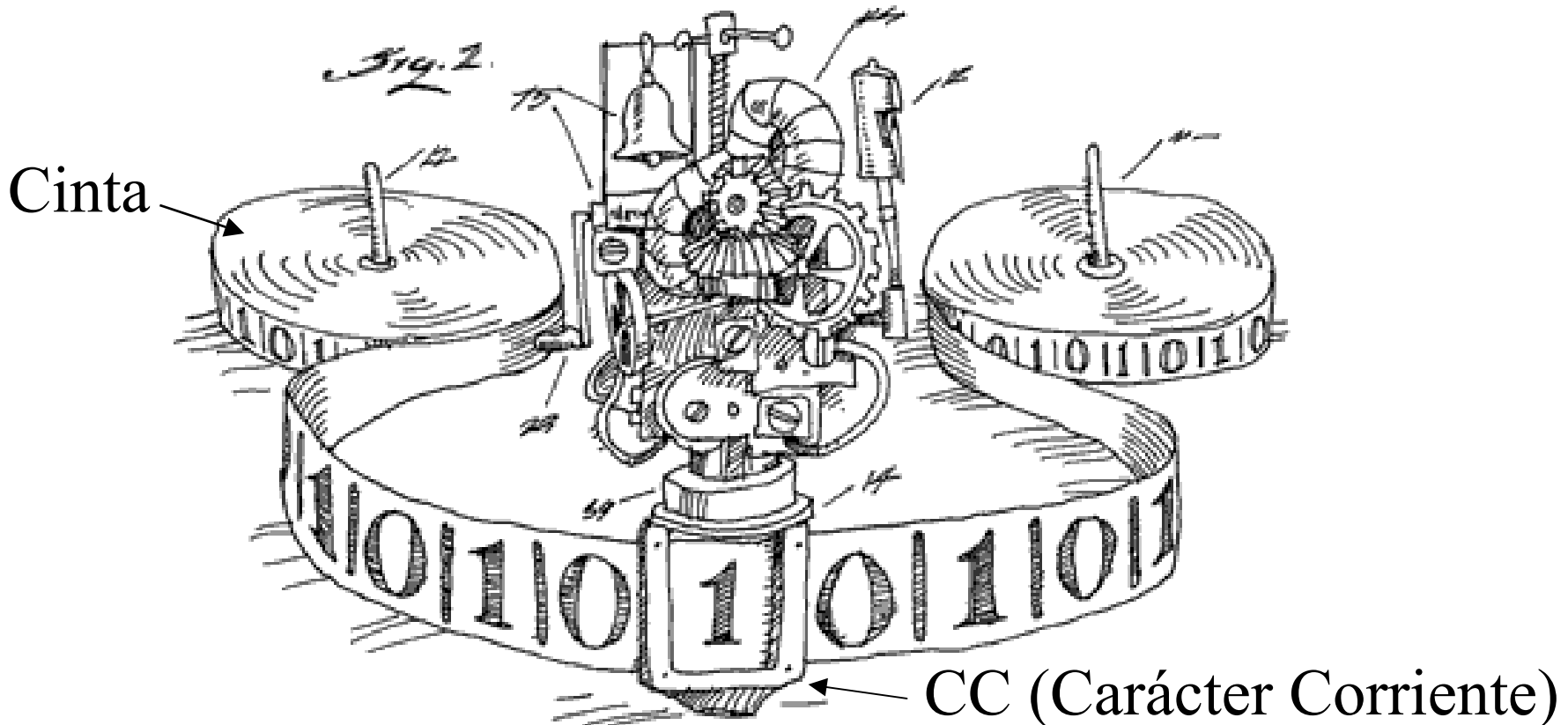
Máquina de Adquisición de Caracteres (MAC)

Es una caja negra con los siguientes componentes:

- Una cinta que contiene una secuencia de caracteres
- Dos botones ARRANCAR (ARR) y AVANZAR (AV)
- Una ventana llamada CARÁCTER_CORRIENTE (CC)

Máquina de Adquisición de Caracteres (MAC)

^Físicamente podría ser así:



Máquina de Adquisición de Caracteres (MAC)

Utilización: consiste en pasar una cinta a través de la ventana. Este desplazamiento se hace siempre desde el inicio y controlado por 2 botones.

La cinta esta compuesta por secuencia de casillas, cada casilla contiene un carácter. Se hace abstracción de la representación del carácter dentro de la casilla.

Máquina de Adquisición de Caracteres (MAC)

La secuencia termina con un carácter especial llamado “carácter corriente” por ejemplo ‘.’ (el punto).

Para comenzar es necesario usar el botón ARR para posicionar el primer carácter de la cinta en la ventana.

El acceso a los demás caracteres se hace mediante el botón AV.

Máquina de Adquisición de Caracteres (MAC)

NOTAS:

1. El cada momento el único carácter de la cinta al cual el usuario tiene acceso es el presente en la ventana CC
2. Toda cinta contiene al menos el carácter terminal
3. No debe presionarse el botón AV cuando el CC sea el carácter terminal

MAC: Primitivas | Operaciones

Acciones que cambian el estado de la máquina:

ACCION ARRANCAR

{Habilita el acceso al primer carácter de la cinta,
abreviación ARR}

EI = {Cualquiera, la cinta contiene al menos el
carácter terminal}

EF = {CC = primer carácter de la cinta, quizás el
carácter terminal}

FINACCION

MAC: Primitivas | Operaciones

Acciones que cambian el estado de la máquina:

ACCION AVANZAR

{Habilita el acceso al próximo carácter de la cinta, abreviación AV}

EI = {CC = X, X no es el carácter terminal}

EF = {CC = Sucesor(X), quizás el carácter terminal}

FINACCION

MAC: Primitivas | Operaciones

Función de consulta (NO cambia el estado de la máquina):

FUNCION CARÁCTER_CORRIENTE

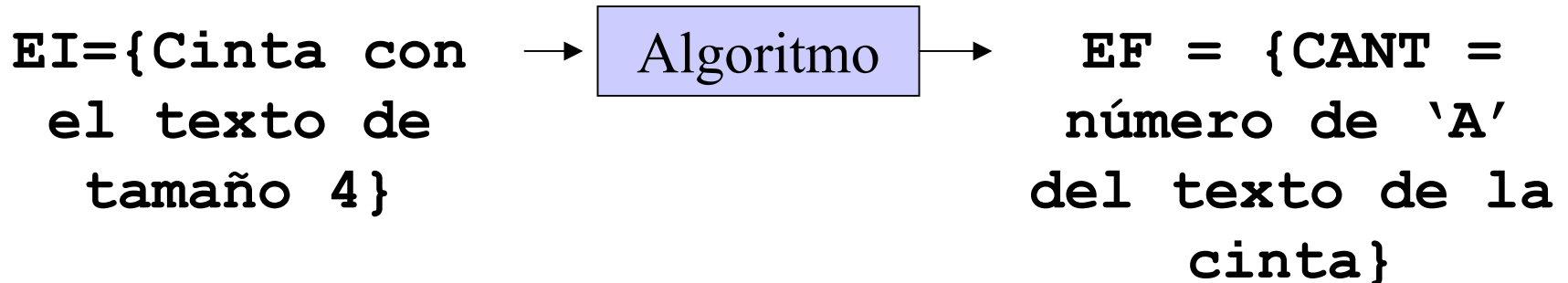
{Da acceso al carácter de la secuencia presente en la ventana, abreviación CC}

FINFUNCION

Ejemplo 7:

Contar las 'A' de una secuencia

Enunciado: La cinta de una MAC contiene un texto formado por 4 caracteres y terminado en '.'. Se quiere contar la cantidad de letras 'A' de ese texto.



Solución: 1era versión

Estrategia: Se basa en el recorrido de la secuencia de caracteres (usando las primitivas) y el secuenciamiento de las acciones que tratan cada carácter: si el carácter es una 'A' se incrementa el contador de A, si no el carácter es ignorado. El proceso se hace 4 veces

LEXICO:

Entero CANT: contador con la cantidad de letras 'A' del texto

Solución: 1era versión

INICIO

EI = {la MAC contiene un texto de 4 caracteres terminado en '.'}

ARR

CANT \leftarrow 0

SI CC = 'A' ENTONCES CANT \leftarrow CANT + 1 FINSI

AV

SI CC = 'A' ENTONCES CANT \leftarrow CANT + 1 FINSI

AV

SI CC = 'A' ENTONCES CANT \leftarrow CANT + 1 FINSI

AV

SI CC = 'A' ENTONCES CANT \leftarrow CANT + 1 FINSI

AV

EF = {CC='.', CANT = cantidad de letras 'A' del texto}

Escribir CANT

FIN

Solución: 1era versión

Notas:

- El tamaño del algoritmo depende del número de letras del texto
- Para dar una solución al problema con las herramientas disponibles es indispensable conocer el número de caracteres del texto
- Necesitamos formas de composición de acciones que permitan construir algoritmos que traten secuencias independientemente del tamaño de la secuencia → Simplificando y generalizando la solución.

Estructuras de Iteración o Repetición

Un lenguaje de programación debe tener como mínimo alguna manera de repetir acciones y controlar su terminación. Las estructuras de repetición o iteración son:

1. REPETIR
2. MIENTRAS
3. ITERAR

REPETIR

Se usa para el tratamiento de secuencias donde es necesario “repetir” una acción elemental o compuesta un número finito de veces (generalmente no conocido)

Se tiene una condición de terminación que se evalúa después de la ejecución del bloque de acciones a repetir. Las acciones a repetir se ejecutan al menos una vez.

Notación Algorítmica

REPETIR

<Secuencia de acciones>

HASTA <expresión lógica>

Donde <expresión lógica> es un predicado o proposición que caracteriza la terminación del proceso iterativo. <Secuencia de acciones> es la secuencia de acciones que se quiere repetir.

Ejemplo 7 - Solución: 2da versión

Estrategia: Generalización del ejemplo 7. Hay que determinar el conjunto de acciones que se repiten y el predicado que determina o caracteriza la terminación del proceso iterativo.

Las acciones que se repiten:

- Tratamiento de cada carácter
- Acceso a un nuevo carácter de la secuencia

LEXICO:

Entero CANT: contador con la cantidad de
letras 'A' del texto

Solución: 2da versión

INICIO

ARR

CANT \leftarrow 0

REPETIR

INV = {CC diferente a '.', CANT = cantidad de
letras 'A' encontradas antes del CC}

SI CC = 'A' ENTONCES

CANT \leftarrow CANT + 1

FINSI

AV

HASTA CC = '.'

Escribir CANT

FIN

Solución: 2da versión

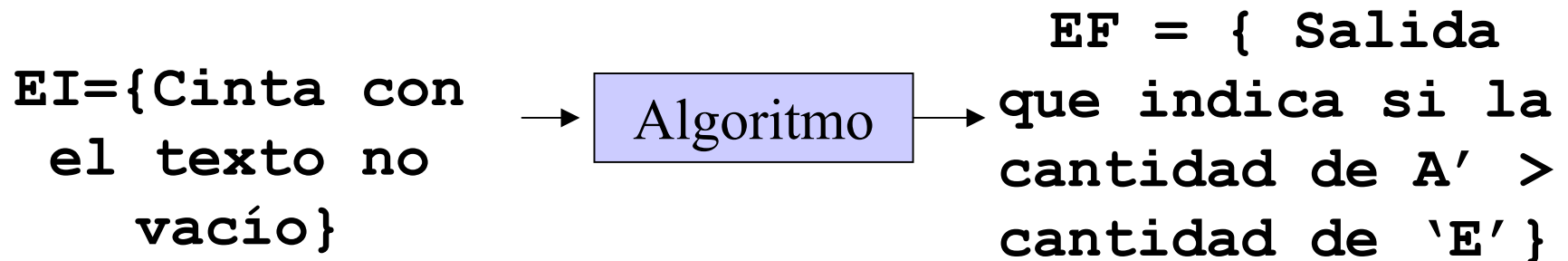
Notas:

- El proceso iterativo termina basándose en la definición de la secuencia (no vacía y terminada en punto) y la especificación de las primitivas de la MAC (ARR, AV y CC)
- La solución es independiente del tamaño de la secuencia → Simplificando y generalizando la solución.
- Se agrega un estado al comienzo de la iteración llamado “invariante” (INV), el cual debe satisfacerse en cada iteración

Ejemplo 8:

Comparación de las 'A' y las 'E'

Enunciado: La cinta de una MAC contiene un texto no vacío terminado en '.'. Se quiere determinar si este texto contiene una cantidad de letras 'A' superior a la cantidad de letras 'E' de ese texto.



Solución: 1era versión

Estrategia: Se basa en el recorrido de la secuencia de caracteres (usando las primitivas) y el secuenciamiento de las acciones que tratan cada carácter: si el carácter es una 'A' se incrementa el contador de A, si no el carácter es ignorado. El proceso se hace 4 veces

LEXICO:

Entero CANTA: contador con la cantidad de letras 'A' del texto

Entero CANTE: contador con la cantidad de letras 'E' del texto

Solución: 1era versión

INICIO

ARR

CANTA \leftarrow 0; CANTE \leftarrow 0

REPETIR

INV = {CC diferente a '.', CANTA = cantidad de letras 'A' encontradas antes del CC, CANTE = cantidad de letras 'E' encontradas antes del CC}

SI CC = 'A' ENTONCES CANTA \leftarrow CANTA + 1 FINSI

SI CC = 'E' ENTONCES CANTE \leftarrow CANTE + 1 FINSI

AV

HASTA CC = '.'

SI (CANTA > CANTE) ENTONCES

Escribir "La cantidad de letras 'A' es MAYOR que la cantidad de letras 'E'"

SINO

Escribir "La cantidad de letras 'A' es MENOR que la cantidad de letras 'E'"

FSI

FIN

Ejemplo 9:

Contar los 'EL'

Enunciado: La cinta de una MAC contiene un texto terminado en '.'. Se quiere contar la cantidad de pares 'EL' de dicho texto.

EI={Cinta con
el texto
terminado en
'.'}



EF = { CANTEL =
cantidad de
pares 'EL' del
texto }

Solución: 1era versión

Estrategia: El texto puede ser vacío. Primero se discrimina el tratamiento del texto vacío y del texto no vacío. Se utiliza una acción nominada llamada `CONTAR_LOS_EL`

Análisis por casos del CC para determinar si es una 'E' o no. En caso de serlo, se avanza al siguiente carácter, incrementando un contador si es una 'L'. En cada iteración se puede tratar 1 o 2 caracteres según el caso.

LEXICO:

Entero `CANTEL`: contador con la cantidad de pares 'EL' del texto

Solución: 1era versión

INICIO

EI = {la cinta de la MAC contiene un texto terminado en '.'}

SI CC = '.' ENTONCES

Escribir "Texto vacío"

SINO

CONTAR_LOS_EL

FINSI

EF = {El texto de entrada fue tratado}

FIN

LEXICO:

ACCION CONTAR_LOS_EL Acción que cuenta e imprime el número de pares 'EL' del texto de la cinta de la MAC

Solución: 1era versión

ACCION CONTAR_LOS_EL

EI = {CC es el primer carácter del texto, CC es diferente de \.'}

CANTEL \leftarrow 0

REPETIR

INV = {CANTEL = cantidad de pares 'EL' encontradas antes del CC}

SI CC = 'E' ENTONCES

AV

SI CC = 'L' ENTONCES

CANTEL \leftarrow CANTEL + 1

FINSI

FINSI

AV

HASTA CC = \.'

EF = {CANTEL es la cantidad de pares 'EL' del texto}

Escribir "La cantidad de pares 'EL' " CANTEL

FINACCION

Solución: 2da versión

Estrategia: Se utiliza una acción nominada llamada
CONTAR_LOS_EL

¿Cómo el CC interviene en la cantidad de pares 'EL' si conocemos la cantidad de pares 'EL' anteriores a este CC?. Se deduce la necesidad de tener un indicador sobre el carácter precedente

LEXICO:

Entero CANTEL: contador con la cantidad de
pares 'EL' del texto

Lógico PE: lógico que indica si el carácter
precedente es una 'E'

Solución: 2da versión CONTAR_LOS_EL

ACCION CONTAR_LOS_EL

CANTEL \leftarrow 0

PE \leftarrow falso

REPETIR

INV = {CANTEL = cantidad de pares 'EL' encontradas antes del CC;
PE indica si el carácter precedente a CC es una 'E'}

SI CC = 'E' ENTONCES PE \leftarrow verdadero FINSI

SI CC = 'L' ENTONCES

SI PE = verdadero ENTONCES

CANTEL \leftarrow CANTEL + 1

PE \leftarrow falso

FINSI

FINSI

SI (CC \neq 'E' y CC \neq 'L') ENTONCES PE \leftarrow falso FINSI

AV

HASTA CC = '.'

Escribir "La cantidad de pares 'EL'" CANTEL

FINACCION

Solución: 3ra versión

Estrategia: Se utiliza una acción nominada llamada
CONTAR_LOS_EL

Difiere de la versión 2 en el tratamiento del carácter corriente (CC):

- Expresión de las condiciones para modificar el contador
- Modificación del indicador PE

LEXICO:

Entero CANTEL: contador con la cantidad de pares 'EL' del texto

Lógico PE: lógico que indica si el carácter precedente es una 'E'

Solución: 3ra versión CONTAR_LOS_EL

ACCION CONTAR_LOS_EL

CANTEL \leftarrow 0

PE \leftarrow falso

REPETIR

INV = {CANTEL = cantidad de pares 'EL' encontradas antes del CC; PE indica si el carácter precedente a CC es una 'E' }

SI (CC = 'L' y PE = verdadero) ENTONCES

CANTEL \leftarrow CANTEL + 1

FINSI

PE \leftarrow (CC = 'E')

AV

HASTA CC = '.'

Escribir "La cantidad de pares 'EL' " CANTEL

FINACCION

Solución: 4ta versión

Estrategia: Se utiliza una acción nominada llamada `CONTAR_LOS_EL`

Reducir el problema al tratamiento de secuencia con pares de caracteres del texto:

- Todo par consta de 2 caracteres consecutivos
- El carácter final de un par cualquiera es el carácter inicial del siguiente par. Excepto el último par cuyo carácter final es un ‘.’

Solución: 4ta versión

Estrategia: tratar pares

Ejemplo: secuencia 'LA CASA.'

- 1er par: 'LA'
- 2do par: 'A_'
- 3er par: '_C'
- 4to. Par: 'CA'
- 5to. Par: 'AS'
- 6to. Par: 'SA'
- 7mo. Par: 'A.'

Solución: 4ta versión

Estrategia: Máquina de Adquisición de pares de Caracteres (MAPC), por analogía a MAC consta de las siguientes primitivas:

- **PAR_CORRIENTE**: da acceso al par de la ventana de la MAPC
- **PAR_TERMINAL**: indica si es el último par de la secuencia

Solución: 4ta versión

Estrategia: Máquina de Adquisición de pares de Caracteres (MAPC), primitivas:

- ARR_PAR: la cinta de al menos un par, muestra en la ventana el primer par, puede ser el par terminal
- AV_PAR: PAR_CORRIENTE = X_1X_2 , no es el par terminal, PAR_CORRIENTE = Sucesor(X_1X_2) quizás es el par terminal

Solución: 4ta versión CONTAR_LOS_EL

ACCION CONTAR_LOS_EL

ARR_PAR

CANTEL \leftarrow 0

SI NO(PAR_TERMINAL) ENTONCES

REPETIR

INV = {CANTEL = cantidad de pares 'EL' encontradas antes del
PAR_CORRIENTE;}

SI (PAR_CORRIENTE = 'EL') ENTONCES

CANTEL \leftarrow CANTEL + 1

FINSI

AV

HASTA PAR_TERMINAL

FINSI

Escribir "La cantidad de pares 'EL' " CANTEL

FINACCION

Solución: 4ta versión CONTAR_LOS_EL

Primitivas de las MAPC:

El PAR_CORRIENTE esta formado por 2 caracteres
PRIMCAR y SEGUNCAR

ACCION ARR_PAR

PRIMCAR ← CC

AV

SEGUNCAR ← CC

FINACCION

ACCION AV_PAR

PRIMCAR ← SEGUNCAR

AV

SEGUNCAR ← CC

FINACCION

FUNCION PAR_TERMINAL : LOGICO

RETORNA (SEGUNCAR = \.')

FINFUNCION

Solución: 4ta versión

En esta solución se eleva el nivel de abstracción al definir la MAPC definida en términos de la MAC. Si sustituimos el algoritmo de cada operación o primitiva de esta nueva máquina en el algoritmo de la acción `CONTAR_LOS_EL`, se obtiene otra versión (5ta) de la solución del problema.

Nótese que el `PAR_CORRIENTE` esta formado por 2 caracteres `PRIMCAR` Y `SEGUNCAR`.

Solución: 5ta versión CONTAR_LOS_EL

ACCION CONTAR_LOS_EL

PRIMCAR \leftarrow CC; AV; SEGUNCAR \leftarrow CC

CANTEL \leftarrow 0

SI SEGUNCAR = '.' ENTONCES

REPETIR

INV = {CANTEL = cantidad de pares 'EL' encontradas antes del
PAR_CORRIENTE;}

SI (PRIMCAR = 'E' Y SEGUNCAR = 'L') ENTONCES

CANTEL \leftarrow CANTEL + 1

FINSI

PRIMCAR \leftarrow SEGUNCAR; AV; SEGUNCAR \leftarrow CC

HASTA SEGUNCAR = '.'

FINSI

Escribir "La cantidad de pares 'EL'" CANTEL

FINACCION

TAREA

A. Revise la implementación en el lenguaje de programación C, de todos los ejemplos y ejercicios de este tema – Estructuras de iteración: repetir. Aclare dudas en el laboratorio o por los canales virtuales con el preparador y/o profesor.

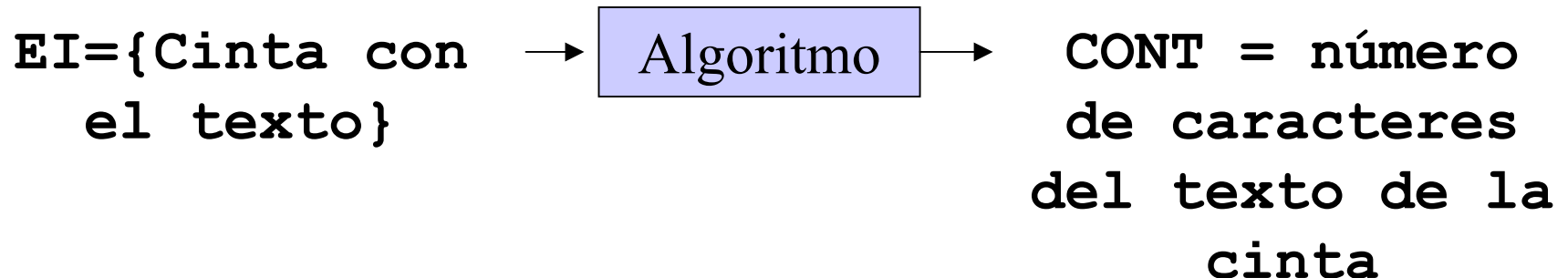
MIENTRAS

Se usa para el tratamiento de secuencias. Esta segunda forma de iteración se presenta como una forma de simplificación de la escritura de la solución de ciertos problemas que son expresados con la herramienta repetir. Tal como se muestra en la versión 2 del ejemplo10

Ejemplo 10:

Contar los caracteres de una secuencia

Enunciado: La cinta de una MAC contiene un texto terminado en '.'. Se quiere contar la cantidad de caracteres del texto de la secuencia (puede ser vacía).



Solución: 1era versión – Utilizando el repetir

Estrategia: Se basa en el recorrido de la secuencia de caracteres hasta el '.', donde el tratamiento del CC consiste en el conteo y el acceso al próximo carácter. Debe considerarse la posibilidad del texto vacío.

LEXICO:

Entero CONT: contador con la cantidad de caracteres del texto

Solución: 1era versión - Repetir

INICIO

ARR

CONT \leftarrow 0

SI CC \neq \.' ENTONCES

REPETIR

INV = {CONT = cantidad de caracteres antes
del CC}

CONT \leftarrow CONT + 1

AV

HASTA CC = \.'

FSI

Escribir CONT

FIN

Solución: 1era versión

Nótese que la entrada al repetir esta condicionada por el predicado $CC \neq \cdot$, para considerar la secuencia vacía. Esta condición es equivalente a la negación de la condición de terminación del repetir. En casos como este se sugiere una simplificación utilizando el mientras (2da versión)

Notación Algorítmica

MIENTRAS <expresión lógica> **HACER**
<Secuencia de acciones>

FINMIENTRAS

Donde <expresión lógica> es un predicado o proposición que caracteriza el comienzo y la permanencia en el proceso iterativo. <Secuencia de acciones> es la secuencia de acciones que se quiere repetir.

Solución: 2da versión - Mientras

INICIO

ARR

CONT \leftarrow 0

MIENTRAS CC \neq '.' HACER

INV = {CONT = cantidad caracteres antes
del CC}

CONT \leftarrow CONT + 1

AV

FINMIENTRAS

Escribir CONT

FIN

MIENTRAS

Notas:

- El test de comienzo o continuación se efectúa **antes** de la ejecución del bloque de acciones. Así en el mientras la secuencia de acciones se ejecutará cero (0) ó mas veces.
- La proposición que caracteriza el comienzo o continuación del mientras es equivalente a la negación de terminación (repetir)

Equivalencia MIENTRAS y REPETIR

Sea α un predicado que caracteriza la terminación del proceso iterativo y A una secuencia de acciones a repetir, la equivalencia Repetir – Mientras es la siguiente:

<u>SI NO (α) ENTONCES</u> <u>REPETIR</u> A <u>HASTA α</u> <u>FINSI</u>	<u>MIENTRAS NO (α) HACER</u> A <u>FINMIENTRAS</u>
--	---

ITERAR

Tercera forma de iteración.

Las 2 formas de iteración anteriores se caracterizan por que el chequeo de la condición de parada de la iteración se realiza:

1. En el **REPETIR** al final
2. En el **MIENTRAS** al comienzo

En el iterar dicho chequeo se puede realizar en cualquier parte del cuerpo de acciones, por lo cual es la herramienta mas general

Notación Algorítmica

ITERAR

A

PARADA α

B

FINITERAR

Sea α una proposición que caracteriza la terminación de un proceso iterativo y sean **A** y **B** 2 secuencias de acciones cualesquiera. Esto indica que **A** se ejecuta 1 o más veces, mientras que **B** se ejecuta 0 o mas veces.

Equivalencias ITERAR

Si la secuencia de acciones **B** es vacía el iterar se comporta como un repetir. Si la secuencia de acciones **A** es vacía el iterar se comporta como un mientras.

<u>REPETIR</u> A <u>HASTA</u> α	<u>ITERAR</u> A <u>PARADA</u> α <u>FINITERAR</u>
---	---

<u>MIENTRAS</u> NO (α) <u>HACER</u> B <u>FINMIENTRAS</u>	<u>ITERAR</u> <u>PARADA</u> α B <u>FINITERAR</u>
--	---

Equivalencias ITERAR

Si las secuencias de acciones **A** y **B** no son vacías el iterar se comporta como un mientras con las siguientes características:

A <u>MIENTRAS</u> NO (α) <u>HACER</u> B A <u>FINMIENTRAS</u>	<u>ITERAR</u> A <u>PARADA</u> α B <u>FINITERAR</u>
---	--

ITERAR

Para los problemas de tratamiento de secuencia donde el fin de la secuencia esta determinado por una propiedad que cumple el último elemento de la misma, el iterar es la herramienta mas adecuada ya que evita la repetición de ciertas acciones tales como: tratamiento del primer carácter y ultimo carácter de la secuencia.

Ejemplo 11:

Existencias de Diptongos en el texto

Enunciado: La cinta de una MAC contiene un texto en Castellano terminado en '.'. Se quiere determinar si dicho texto contiene al menos un diptongo. Un diptongo es la aparición de 2 vocales débiles (i,u) o de una vocal débil y una vocal fuerte (a,e,o) consecutivamente en una palabra

EI={Cinta con
el texto en
Castellano}



EF = { Lógico
ENC indica si
hay un diptongo
en el texto}

Solución: 1era versión

Estrategia: Tratamiento del carácter corriente consiste en determinar si es una vocal débil o fuerte o cualquier otro carácter. En caso de ser una vocal débil se analiza si el CC anterior fue otra vocal débil, para esto se mantiene la información sobre el carácter precedente (indicador PVD), si es así se puede terminar con el proceso iterativo, sino se actualiza PVD y se continúa el proceso.

Solución: 1era versión

Estrategia: En caso de que CC sea una vocal fuerte basta con verificar si PVD es verdadero para terminar el proceso iterativo con éxito, sino se continua la búsqueda. En cualquier otro caso se actualiza PDV.

LEXICO:

Lógico PVD: indica si el carácter anterior al CC es una vocal débil

Lógico ENC: indica el éxito o el fracaso del proceso de búsqueda de diptongos

Solución: 1era versión

INICIO

EI = {La cinta de una MAC contiene un texto en Castellano terminado en punto}

ARR

SI (CC = '.') ENTONCES

Escribir "El texto es vacío"

SINO

PVD ← falso; ENC ← falso

ITERAR

INV = {PVD = indica si el carácter precedente a CC es una vocal debil}

SI (CC = 'I') o (CC = 'U') ENTONCES

SI PVD ENTONCES ENC ← verdadero

SINO PVD ← verdadero FINSI

FINSI

SI (CC = 'A') o (CC = 'E') o (CC = 'O') ENTONCES

SI PVD ENTONCES ENC ← verdadero FINSI

FINSI

SI (CC != 'A') y (CC != 'E') y (CC != 'I') y (CC != 'O') y (CC != 'U') ENTONCES PDV ← falso FSI

Solución: 1era versión

HASTA ENC o (CC = \.')

AV

FINITERAR

SI ENC ENTONCES

Escribir "El texto contiene diptongos"

SINO

Escribir "El texto NO contiene diptongos"

FINSI

FINSI

EF = {El texto ha sido tratado y se muestra si hay o
no un diptongo}

FIN

TAREA

- A. Revise la implementación en el lenguaje de programación C, de todos los ejemplos y ejercicios de este tema – Estructuras de iteración: iterar y mientras. Aclare dudas en el laboratorio o por los canales virtuales con el preparador y/o profesor.**

Ejercicios propuestos

1. Elabore un algoritmo para resolver cada uno de los siguientes problemas donde la cinta de una MAC contiene un texto terminado en punto '.', se quiere:
 - a) Contar la cantidad de caracteres distintos a 'A'
 - b) Contar la cantidad de vocales y de consonantes
 - c) Calcular el porcentaje de letras 'A' con respecto al total de caracteres distintos de espacio
 - d) Generalice los ejercicios (a) y (c) al considerar un carácter de entrada cualquiera
 - e) Verifique si existe la segunda aparición del primer carácter del texto. Si existe, indicar la posición que ocupa.

Ejercicios propuestos

2. Modifique las 4 versiones del ejemplo 9, considerando que se quiere:
 - a) Contar los 'EL' y los 'ES'
 - b) Contar los 'EL' y los 'UN'
 - c) Contar la cantidad de pares de caracteres iguales a un par de entrada cualquiera
3. Basado en el ejercicio anterior analice la legibilidad y adaptabilidad de las 4 versiones del ejemplo 10. Cuál le parece más legible? Cuál más adaptable?

Ejercicios propuestos

4. Elabore un algoritmo para resolver cada uno de los siguientes problemas para una cinta de una MAC que contiene un texto terminado en punto '.', se quiere:
 - a) Contar las palabras del texto
 - b) Contar la cantidad de palabras del texto que estén formadas por una sola letra
 - c) Contar la cantidad de palabras del texto que estén formadas por dos (2) letras
 - d) Contar la cantidad de palabras del texto que estén formadas por tres (3) letras
 - e) Contar la cantidad de casos en que una palabra comienza con el mismo carácter con que termina

Ejercicios propuestos

- a) Contar la cantidad de casos en que una palabra termina con el mismo carácter con el que comienza la siguiente
 - b) Contar la cantidad de diptongos que hay en el texto. Asuma que el texto está en Castellano
 - c) Contar la cantidad de triptongos que hay en el texto. Asuma que el texto está en Castellano
5. La cinta de una MAC contiene un texto finalizado en ‘.’. Se quiere elaborar un algoritmo el cual dado dos (2) letras del alfabeto nominadas L1, L2, determine si existe alguna palabra en el texto que posea estas 2 letras (no necesariamente contiguas) en el orden L1, L2.

Máquina de Creación de Secuencias de Caracteres (MCS)

Con el objetivo de completar el tratamiento de secuencias, se define una máquina hipotética, haciendo abstracción de la representación de la secuencia de caracteres, que permite que los algoritmos generen como resultado una secuencia de caracteres.

Máquina de Creación de Secuencias de Caracteres (MCS)

Es una caja negra análoga a la MAC con los siguientes componentes:

- Una cinta para grabar los caracteres de la secuencia
- Dos primitivas: PREPARAR (PREP) y GRABAR (GRAB)

Máquina de Creación de Secuencia de Caracteres (MCS)

Para la utilización de la MCS es necesario:

- Habilitar la cinta para la grabación mediante la primitiva PREP.
- Grabar los caracteres que correspondan mediante la primitiva GRAB.
- Grabar un carácter especial al final de la secuencia (carácter terminal), por ejemplo ‘.’.
- El orden de grabación de los caracteres impone el orden de los caracteres en la secuencia creada

MCS: Primitivas | Operaciones

Acciones que cambian el estado de la máquina:

ACCION PREPARAR

{Habilita la cinta para la creación de la nueva secuencia, abreviación PREP}

EI = {Cualquiera, la máquina contiene una cinta}

EF = {La máquina está lista para grabar el primer carácter}

FINACCION

MCS: Primitivas | Operaciones

Acciones que cambian el estado de la máquina:

ACCION GRABAR (Caracter C)

{Graba el caracter nominado por C a continuación de los caracteres ya grabados, el valor nominado por C no se altera. Abreviación GRAB(C)}

EI = {N caracteres están grabados en la cinta, $N \geq 0$ }

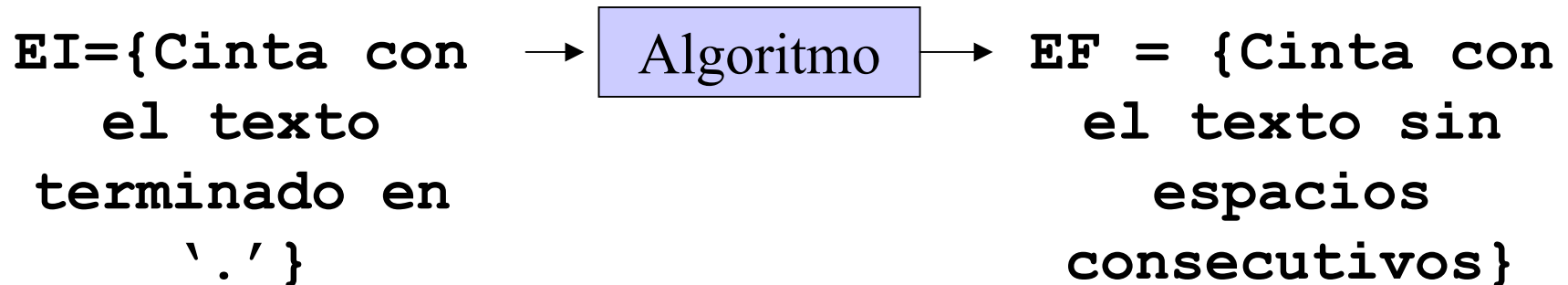
EF = {El carácter nominado por C se grabó a continuación de los N primeros caracteres de la cinta}

FINACCION

Ejemplo 12:

Reducción de los espacios

Enunciado: La cinta de una MAC contiene un texto terminado en '.'. Se quiere grabar sobre otra cinta el texto de entrada reduciendo cada grupo de espacios por un solo espacio.



Solución: 1era versión

Estrategia: Tratamiento sobre la secuencia de entrada depende de si el CC es una letra o un espacio:

- Si el CC es una letra se debe grabar
- Si el CC es un espacio se debe grabar siempre y cuando sea el primero de un grupo de espacios. Para eso se guarda la información del carácter precedente al CC si es o no un espacio a través de un lógico PE

LEXICO:

Lógico PE: indica si el carácter anterior a CC es un espacio

Solución: 1era versión

INICIO

PREP

ARR

PE ← falso

MIENTRAS CC != \.' HACER

INV = {PE indica si el precedente a CC es un espacio, todos los caracteres a grabar antes de CC se han grabado}

SI (CC != \ '\) o no(PE) ENTONCES

GRAB(CC)

FINSI

PE ← (CC = \ '\)

AV

FINMIENTRAS

GRAB(\.')

FIN

Solución: 2da versión

Estrategia: Tratamiento sobre la secuencia de salida. Esta versión define una MAC1 que da acceso a los caracteres de la secuencia a grabar a partir de la secuencia de entrada. Dicha máquina es análoga a la MAC y tiene las siguientes primitivas:

- **ARR1**: habilita el acceso al primer carácter de la secuencia de salida

Solución: 2da versión

- **AV1:** Habilita el acceso al próximo carácter de la secuencia de salida. Si $CC1 = X$, al avanzar se retorna el sucesor de X en la secuencia de salida
- **CC1:** da acceso al carácter actual de la secuencia de salida, ventana de la MAC1.

Así, el tratamiento del CC de la secuencia de salida (CC1) consiste simplemente en su grabación tal como lo muestra el siguiente algoritmo

Solución: 2da versión

INICIO

PREP

ARR1

MIENTRAS CC1 != \.' HACER

INV = {Todos los caracteres anteriores a
CC1 se han grabado}

GRAB (CC1)

AV1

FINMIENTRAS

GRAB (\.')

FIN

Solución: 2da versión

Primitivas de las MAC1:

ACCION ARR1

{Da acceso al primer carácter de la secuencia de salida }

EI = {Cualquiera}

AV

CC1 ← CC

EF = {CC1 es el primer carácter de la secuencia de salida}

FINACCION

El primer carácter de la secuencia de entrada es el primero de la secuencia de salida

Solución: 2da versión

Esta acción determina el valor de CC con las siguientes reglas:

1. Si el CC de la secuencia de entrada es una letra entonces el próximo de la secuencia de salida será el siguiente carácter a CC
2. Si el CC de la secuencia de entrada es un espacio entonces el próximo de la secuencia de salida será el próximo carácter distinto de espacio de la secuencia de entrada

ACCION AV1

SI CC != ' ' ENTONCES AV

SINO

REPETIR

AV

HASTA (CC != ' ')

FINSI

CC1 ← CC

FINACCION

Solución: 2da versión

FUNCION CC1

Esta conformada por el nombre CC1 que denota el carácter corriente a grabar en el texto de salida

FINFUNCION

Nota: Se puede sustituir el cuerpo de las acciones de la MAC1 de la secuencia de salida en el algoritmo segunda versión (ARR1, AV1), para colocarla en términos de la entrada solamente. Las acciones $CC1 \leftarrow CC$ puede eliminarse si se sustituye CC1 por CC. La tercera versión de la solución se muestra en el siguiente algoritmo

Solución: 3ra versión

INICIO

PREP

AV

MIENTRAS CC != \.' HACER

INV = {Todos los caracteres anteriores a CC se
han grabado}

GRAB (CC)

SI CC != \ \ ENTONCES AV

SINO

REPETIR

AV

HASTA (CC != \ \)

FINSI

FINMIENTRAS

GRAB (\.')

FIN

Ejemplo 13:

Truncamiento de palabras

Enunciado: La cinta de una MAC contiene un texto formada por una secuencia de palabras separadas por al menos un espacio y terminado en ‘.’. Se quiere grabar en otra cinta el texto de entrada con la siguiente restricción:

- Si el tamaño de la palabra es menor o igual que cierto valor K , la palabra se graba completa
- Si el tamaño de la palabra es mayor que k , solo se graban los primeros K caracteres de la palabra
- Se asume que K es mayor o igual a 1

Ejemplo 13:

Truncamiento de palabras

Estrategia: El problema consiste en definir la información necesaria para determinar cuando un carácter de la secuencia de entrada debe aparecer en el texto de Salida. Se tiene un identificador NOR que indica el número de orden del carácter anterior al CC dentro de cada palabra del texto. Si el carácter anterior al CC es un espacio, dicho identificador tendrá el valor cero.

**EI={Cinta con
secuencia de
palabras, K
entero
positivo}**



**EF = { Cinta con
secuencia de
palabras de
tamaño menor o
igual a K}**

Solución: 1era versión

INICIO

Leer K

EI = {La cinta de una MAC contiene un texto en terminado en punto. $K = k$ y $k > 1$ }

PREP; ARR

NOR \leftarrow 0

MIENTRAS (CC \neq \'.') HACER

INV = {Se han grabado todos los caracteres anteriores a CC según: Si NOR = 0 el carácter anterior es un blanco. Si NOR > 0 el carácter anterior a CC es el carácter que esta en la posición NOR dentro de la palabra que contiene a ese caracter}

SI (CC = \ ') ENTONCES

GRAB (CC)

NOR \leftarrow 0

SINO

SI (NOR = 0) ENTONCES

GRAB (CC)

NOR \leftarrow 1

Solución: 1era versión

```
SINO
  SI (NOR < K) ENTONCES
    GRAB (CC)
    NOR ← NOR + 1
  SINO
    SI (NOR >= K) ENTONCES
      NOR ← NOR + 1
    FINSI
  FSI
FSI
FINSI
AV
FINMIENTRAS
GRAB ( \ . ' )
```

EI = {El texto de salida fue creado acorde a las condiciones dadas}

FIN
LEXICO
Entero NOR: indica el número de orden del carácter anterior al CC dentro de cada palabra

Ejercicios propuestos

1. La cinta de una MAC contiene un texto formado por palabras y espacio, terminado en punto '.', se quiere:
 - a) Transformar el texto dado, intercambiando los caracteres 2 a 2, si la cantidad de caracteres del texto es impar, el último carácter no se intercambia. Por ejemplo:

Texto de entrada: 'DIME CON QUIEN ANDAS'

Texto de salida: 'IDEMC NOQ IUNEA DNSA'
 - b) Transformar el texto dado, suprimir todas aquellas palabras que sean de 1 o 2 caracteres
 - c) Grabar sobre otra cinta el mismo texto de entrada con las siguientes condiciones:
 - Reducir todo grupo de espacios a uno solo
 - Suprimir los espacios eventuales que estén antes de la primera palabra y después de la última

Ejercicios propuestos

- d) Dado un valor entero $K(K>0)$ se quiere producir un nuevo texto, eliminando del texto de entrada a todas aquellas palabras que estén en una posición múltiplo de K .
2. Elabore un algoritmo para resolver el siguiente problema: La cinta de una MAC tiene un texto terminado en `'.'`. Dicho texto es una secuencia de frases terminadas en `‘;’` y una frase es una secuencia de palabras separadas por blancos. Las frases pueden ser vacías (solo blancos o nada). Se quiere:
- Indicar el número de palabras que hay en cada frase del texto
 - Producir un nuevo texto de acuerdo a las siguientes condiciones:
 - Todo grupo de espacios debe reducirse a uno solo
 - Toda frase vacía se debe suprimir

"Somos lo que hacemos
repetidamente.

La excelencia
entonces no es
un acto, sino un
hábito."

- Aristoteles

