SOFTWARE, INSTRUMENTACION Y METODOLOGÍA

FRA3: UN PROGRAMA EN PASCAL PARA LA INVESTIGACIÓN SOBRE LOS PROCESOS DE COMPRENSIÓN EN LA LECTURA

Ramón López Sánchez
Departamento de Psicología Básica II: Procesos Cognitivos. Universidad Complutense de Madrid

FRA3 puede considerarse como una aplicación (en lenguaje Pascal) de una técnica de lectura autoespaciada no-acumulativa que resulta especialmente útil en la investigación sobre el proceso de comprensión lectora. FRA3 presenta el material verbal dividido en fragmentos. También puede incorporar una tarea de decisión léxica en un punto crítico de estos materiales. El fichero de datos de cada sujeto incluye los tiempos de lectura de cada uno de los fragmentos y el tiempo de la decisión léxica. El código fuente se ha dividido en procedimientos para facilitar la descripción de FRA3.

Palabras clave: lenguaje Pascal, lectura autoespaciada no acumulativa, decisión léxica, material verbal, tiempos de lectura, tiempo de decisión.

FRA3: A Pascal application for reading-comprehension research. FRA3 could be considered as an application (in Pascal language) of a no-cumulative self-paced reading task that is specially useful in reading-comprehension process research. FRA3 displays verbal material divided in fragments. It can incorporate a lexical decision task in a critical point of these materials too. Each subject’s data file includes reading times for each of the fragments and the lexical decision time. The program’s original code has been divided in procedures to facilitate FRA3 description.

Key words: Pascal language, no-accumulative self-paced reading task, lexical decision, verbal materials, reading times, decision time.

FRA3 se diseñó con el propósito de utilizar una técnica de lectura auto-espaciada no-acumulativa en las investigaciones sobre el procesamiento de frases localmente ambiguas en la lectura (Rayner et al., 1989; López Sánchez, 1992). Los enunciados ambiguos funcionan como unidades neutralizadas que permiten estudiar la fuerza de los factores que condicionan la desambigüedad (Mayor, 1979, 1980; Mayor y Moya, 1991). Los estudios realizados en este ámbito han empleado generalmente frases que contienen una ambigüedad sintáctica local (frases que son ambiguas hasta un punto pero que incluyen información posterior que las desambigua). Los estudios sobre el procesamiento de la ambigüedad sintáctica han puesto de manifiesto lo que hacen los lectores ante una situación de incertidumbre, y han proporcionado distintas respuestas, desde distintas posiciones teóricas, en relación con tres cuestiones críticas sobre la arquitectura funcional del analizador del lenguaje natural: la estructura del procesador, la disposición temporal de los procesos, y el modo de comunicación entre los componentes del sistema (Frazier, 1987; Taraban y McClelland, 1988; Steedman y Altmann, 1989). En general las hipótesis sobre la arquitectura funcional del procesador se pueden reducir a dos: la concepción modular (Forster, 1979; Fodor,

La representación final de una frase o de un texto es insuficiente para conocer la forma en que se construye la estructura y el significado de una frase palabra a palabra. Por ello se han desarrollado una gran cantidad de técnicas para explorar los procesos que tienen lugar durante la lectura. Una de estas tareas es la lectura auto-espaciada palabra a palabra, en la que los sujetos presionan un botón para leer cada nueva palabra (Mitchell y Green, 1978; Just, et al., 1982; Ferreira y Henderson, 1990). La presentación de las frases en esta modalidad puede ser acumulativa (las palabras se van acumulando sucesivamente en la pantalla a medida que los sujetos pulsan el botón) o no acumulativa (las que ya han sido presentadas desaparecen de la pantalla). Otras modalidades de lectura auto-espaciada pueden incluir la presentación de unidades más largas de un texto, como una cláusula o una frase. Una variedad de esta técnica es la que combina la lectura auto-espaciada con una tarea secundaria, en la que los sujetos leen un fragmento hasta una palabra y en este punto deben denominar la palabra o tomar una decisión léxica sobre la misma (Clifton, Frazier y Connine, 1984; Marslen-Wilson, Brown y Tyler, 1988). La lectura auto-espaciada acumulativa ha sido frecuentemente criticada puesto que los sujetos adoptan estrategias específicas de la tarea (por ejemplo, presentar rápidamente la frase y leerla posteriormente una vez que están disponibles todas las palabras) (Just et al., 1982; Ferreira y Henderson, 1990). Los resultados más próximos a los obtenidos con el registro de los movimientos oculares, que hoy por hoy resultan los más completos y precisos (Ferreira y Henderson, 1990), se han observado utilizando la técnica de lectura auto-espaciada no acumulativa palabra a palabra y la RSVP (Just, Carpenter y Woolley, 1982; Ferreira y Clifton, 1986; Ferreira y Henderson, 1990).

Existen en el mercado algunos programas y paquetes de programas que permiten la implementación de las técnicas de lectura auto-espaciada acumulativa y no acumulativa. Algunos de estos son el MEL (Micro Experimental Laboratory) desarrollado por Psychology Software Tools Inc. y el Mouselab System Program (Johnson et al., 1989). El último programa presenta algunos problemas puesto que los sujetos deben desplazarse a través del texto utilizando un ratón, y lógicamente, la habilidad de los mismos en su manejo puede influir claramente en los resultados. El MEL es un paquete integrado de programas que permite implementar una variedad de tareas clásicas para la investigación en Psicología Cognitiva. Sin embargo, a parte de su coste económico, resulta difícil establecer los parámetros del diseño que permiten llevar a cabo un experimento particular, y no presenta la posibilidad de incluir una tarea secundaria en un punto de las frases.

La aplicación desarrollada en este caso es un caso particular de la técnica de lectura auto-espaciada cláusula a cláusula, donde se presenta una frase básica (dos primeros fragmentos) que puede tener tres terminaciones posibles (tercer fragmento). El programa ofrece también la posibilidad de incluir una tarea de decisión léxica en el punto donde aparece la ambigüedad local de las frases (después del segundo fragmento). La estructuración del programa en procedimientos, hace que FRA3 pueda adaptarse fácilmente a las necesidades de investigación de cada usuario, y también permite una fácil adaptación a cualquiera de las modalidades de lectura auto-espaciada.

I. DESCRIPCION DEL PROGRAMA

La compilación del código fuente que aparece descrito en este apartado se realizó con el Turbo-Pascal v. 3.01.

Los comentarios en algunas líneas del programa aparecen encerrados entre asterís
Figura 1: Diagrama de flujo que describe el funcionamiento general de FRA3.
cos y paréntesis, respetando así el modo de aparición de los mismos en Pascal. Tengase en cuenta que el orden de los procedimientos ha sido alterado para conseguir una mayor claridad expositiva y para ajustarse al diagrama de funcionamiento general del programa.

Los procedimientos que aparecen tras la definición de las variables ocuparían de forma obligatoria la última parte del programa en su versión ejecutable.

La Figura 1 muestra un diagrama de flujo del funcionamiento de FRA3.

1.1.—DEFINICIÓN DE LAS VARIABLES Y SUS TIPOS.

A continuación aparecen la definición de los tipos y las variables utilizadas por el programa junto con anotaciones aclaratorias para identificarlas de una forma más precisa.

Program Frases;
Type
tiempo = record
  seg : integer; (* segundos *)
  cent: integer; (* centésimas de segundo *)
  end;
número = record
  fra : integer; (* Número de frase (1-24) *)
  term: integer; (* Número de terminación (1,2,3) *)
  end;
reg = record
  c_s : char; (* 1-Con palabras, 2-Sin palabras *)
  num : array [1..72] of numero; (* N° de frase y terminac. *)
  med1 : array [1..72] of tiempo;
  med2 : array [1..72] of tiempo;
  med3 : array [1..72] of tiempo;
  med4 : array [1..72] of tiempo;
  bien: integer;
  mal: integer;
  end;
var
datos:file of reg;
aux:reg;
T1: array [1..24] of string[49]; (* 1ª parte de cada frase *)
T2: array [1..24] of string[28]; (* 2ª *)
T3: array [1..24,1..3] of string[30];(* 3ª *)

E1: array [1..4] of string[49]; (* 1ª parte ejemplos *)
E2: array [1..4] of string[28]; (* 2ª *)
E3: array [1..4] of string[30];(* 3ª *)
PALABRAS:array [1..24,1..3] of string[9];(*Palabras de cada frase*)
T: array [1..4] of string[9];(* Palabras que salen en ejemplos *)
SUCESO: array [1..24,1..3] of integer;(* Control de las que han salido *)
con,ok: boolean;
cs,correcto, palabra, resp: char;
x,y,h1,h2,h3,h4,h,m1,m2,m3,m4,m,s1,s2,s3,s4,s,s1,d1,d2,d3,d4,d: integer;
h5,m5,s5,d5,h6,m6,s6,d6: integer;

1.2.—PROCEDIMIENTOS PRINCIPALES.

1.2.1.—Inicio del programa.

A continuación aparece el cuerpo del programa, que hace una llamada al procedimiento no_existe_fichero para crear el fichero de datos, si es que este no existe. Por el contrario, si el fichero de datos ya existe llama al procedimiento entrar_en_programa.

Begin
textmode(bw80);
randomize;
textcolor(9);
assign (datos,'datos.dat');
(*$1-*reset(datos);$1+*);
ok:=(iresult=0);
if not ok then no_existe_fichero;
(*$1-*reset(datos);$1+*);
ok:=(iresult=0);
if ok then entrar_en_programa;
End.

1.2.2.—Creación del fichero de datos si no existe.

Si el fichero de datos no existe todavía este procedimiento presenta un mensaje al usuario para ver si desea crearlo. Si la respuesta es afirmativa crea el fichero donde se almacenaran todos los datos que se recojan con el programa.

Procedure no_existe_fichero;
var creac:char;
begin
  clrscr;textcolor(15);
sound(2000);delay(40);nosound;
gotoxy(30,9);write(‘A T E N C I O N ! ’);
gotoxy(20,12);write(‘El fichero de datos no ha sido en- 
contrado’);
gotoxy(27,14);write(‘Desea crearlo (S/N) ’);
repeat
  gotoxy(49,14);write( ‘ ‘);
  buflen:=1;
  gotoxy(49,14);read(kbd, crea);
  if (upcase(crea)<>'S') and (upcase(crea)<>'N') then
    begin
      sound(2000);delay(50);nosound;
      end;
  until (upcase(crea)=>'S') or (upcase(crea)=>'N');
  if (upcase(crea)=>'S') then
    begin
      gotoxy(27,17);write(‘creando fichero...’);
      rewrite(datos);
      end
  else clrscr;
  textcolor(9);
end;

1.2.4.—Inicialización y carga de los
segmentos de las frases.

En este apartado se incluyen dos procedi- 
imentos, inicializar y cargar_textos. El 
primer procedimiento inicializa (a) las tablas 
 correspondientes a los dos primeros seg- 
mentos de las frases experimentales, (b) la 
matriz que almacena la ocurrencia o no ocu-
renza de cada una de las combinaciones de 
las frases básicas más la terminación, y (c) la 
matriz de terminaciones que correspon- 
den a cada una de las frases experimentales. 
El procedimiento cargar_textos asigna 
cadenas de caracteres (1) a cada una de las ta-
blas creadas para los dos primeros segmen-
tos de las frases experimentales, (2) a la 
matriz de combinaciones de las frases bási-
cas más tipo de terminación, y (3) a las ta-
blas de las frases que aparecen en los ejem-
plos y a los estímulos que aparecen en la 
tarea de decisión léxica de los mismos. En 
este último procedimiento se presentan só-
lo cuatro ejemplos de las frases utilizadas 
para no alargarlo en exceso.

Procedure inicializar;
  var i,j:integer;
  begin
    for i:=1 to 24 do
      begin
        T1[i]:='; 
        T2[i]:='; 
        for j:=1 to 3 do
          begin
            SUCESO[i,j]:=0; 
            T3[i,j]:='; 
          end;
      end;
  end;
Procedure cargar_textos;
begin
  (** Primera parte de cada frase ***)
  T1[1]:='El reportero fotografió al chofer del capi-
tán';
  T1[2]:='El artista pintó a la mula del campesino';
  T1[3]:='El perro ladraba al gato del fiscal';
  
Psicotherma, 1994
1.2.5.—Con o sin decisión léxica.

El procedimiento con_sin permite al usuario realizar un experimento de lectura auto-espaciada que incluye (opción 1) o no incluye (opción 2) una tarea de decisión léxica.

Procedure con_sin;
begin
clrscr;
gotoxy(30,8);write('1 - Con palabras');
gotoxy(30,10);write('2 - Sin palabras');
gotoxy(30,22);write('Pulse opción (1-2) :');
repeat
  cs:= ' ';  
  gotoxy(51,22);write(' ');  
  buflen:=1;  
  gotoxy(51,22);(*$I-*)read(kbd,cs)(*$I+*);  
  ok:=(iresult=0);  
  until (ok=true) and ((cs='1') or (cs='2'));
  con:=false;
  if cs='1' then con:=true;
  if con then cargar_palabras;
  aux.c_s:=cs;
end;

Si el usuario elige la opción 1 (con palabras) el paso siguiente sería cargar la matriz de estímulos de la tarea de decisión léxica con las palabras y no palabras correspondientes a las combinaciones frase básica más terminación. Obsérvese que la disposición de las palabras y las no palabras responde a una regla especial (Ver apartado 1.3.5).

Procedure cargar_palabras;
begin
PALABRAS[1,1]:='CONDUCIR';
PALABRAS[1,2]:='RALIFSED';
PALABRAS[2,1]:='RAECOC';
PALABRAS[2,2]:='LABRAR';
PALABRAS[3,1]:='ARANAR';
PALABRAS[3,2]:='RASUAC';
T[24]:='RAETIELP';
T[24]:='ASESINAR';
T[13]:='BENDECIR';
PALABRAS[2,3]:='RAJROF';
PALABRAS[3,3]:='AMASAR';

PALABRAS[24,3]:='RACILBUP';
end;

1.2.6.- Proceso para la presentación de los ejemplos.

Este proceso es básicamente igual al que tiene lugar para las frases experimentales (la única diferencia es que no se miden los tiempos de lectura y decisión léxica), por lo que se remite al lector a la descripción que aparece en el siguiente apartado.

Procedure ejemplos;
var i,p:integer;
tecla:char;
begin
clrscr;
gotoxy(20,12);textcolor(15);
write('Pulse una tecla para comenzar los ejemplos...');
textcolor(9);
read(kbd,tecla);
for i:=1 to 4 do
begin
clrscr;
gotoxy(15,6);textcolor(15);write('
');
pulsar_tecla(16,6);
gotoxy(15,6);write('E[i]');
p:=length('E[i]');
gotoxy(p+16,6);textcolor(15);write(' 
');
pulsar_tecla(p+16,6);
gotoxy(p+16,6);write(' 
');
gotoxy(p+16,6);write('"
');
pulsar_tecla(p+17,8);
gotoxy(p+16,8);write('"
');
end;
gotoxy(p+16,8);
write('E[i]');
p:=p+length('E[i]');
gotoxy(p+17,8);textcolor(15);write('"');
pulsar_tecla(p+18,8);
end;
Procedure sacar_ejemplo(i:integer);
begin
  gotoxy(35,12);sound(2000);delay(50);nosound;delay(50);
textcolor(15);
write('T[i]');
delay(100);gotoxy(35,12);textcolor(9);write(' 
');
end;

1.2.7.- Proceso para la presentación los ensayos experimentales.

El diagrama de flujo presentado anteriormente describe, bajo el encabezamiento «PROCESO PRINCIPAL», la secuencia de procesos que se realizan dentro de este procedimiento (Ver Figura 1). El procedimiento proceso hace llamadas a una serie de rutinas secundarias que aparecerán bajo el apartado 1.3 de este artículo.

Cuando comienza la presentación del material experimental el programa sortea un número de frase y un número de terminación (Véase apartado 1.3.1). La primera vez que el sujeto pulsa la tecla definida por el procedimiento secundario pulsar_tecla (cursor_hacia-abajo) se toma el tiempo de relaj interno (Ver apartado 1.3.3) y después, aparece el primer fragmento de las frases. Cuando el sujeto pulsa por segunda vez esta tecla, desaparece el primer fragmento, se toma otra vez el tiempo de relaj interno, se calcula la diferencia entre los dos tiempos mencionados (Véase el apartado 1.3.4), y se presenta el segundo fragmento.

Cuando la opción elegida es la 2 (sin decisión léxica) pulsar de nuevo la tecla mencionada causa que desaparezca el segundo fragmento, se toma otra vez el tiempo y se calcula la diferencia entre este tiempo y el anterior, y por último, se presenta el tercer
fragmento. Pulsando otra vez la tecla definida previamente desaparece el tercer fragmento, se toma de nuevo el tiempo, se calcula una nueva diferencia entre los tiempos y, comienza un nuevo ensayo (aparece en la pantalla un asterisco).

Si se ha elegido la opción que incluye la tarea de decisión léxica, después del segundo fragmento aparece un estímulo durante 150 milisegundos (Ver apartado 1.3.5) y, el sujeto debe decidir si este es una palabra o una no-palabra (Véase el apartado 1.3.6). Cuando el sujeto da la respuesta se toma el tiempo del reloj interno y se calcula la diferencia entre el tiempo anterior y el presente. Después de la respuesta aparece de forma inmediata un asterisco en la pantalla, cuando el sujeto pulsa de nuevo la tecla cursor-hacia-abajo se toma el tiempo del reloj interno y aparece el tercer fragmento de las frases (terminación). Por último, cuando vuelve a pulsar esta tecla desaparece el tercer fragmento, se toma de nuevo el tiempo, se calcula la diferencia entre el tiempo previo y el que se ha tomado en última instancia y, aparece el asterisco que señala el inicio de otro ensayo.

Procedure proceso;
var i,p:integer;
tecla:char;
begin
clrscr;
gotoxy(25,12);textcolor(15);
write('Pulse una tecla para comenzar...');textcolor(9);
read(kbd,tecla);
aux:mal:=0;
aux:biern:=0;
for i:=1 to 72 do
begin
clrscr;
aux:med1[i].seg:=0;aux:med1[i].cent:=0;
aux:med2[i].seg:=0;aux:med2[i].cent:=0;
aux:med3[i].seg:=0;aux:med3[i].cent:=0;
aux:med4[i].seg:=0;aux:med4[i].cent:=0;
aux:num[i].fra:=0;
aux:num[i].term:=0;
repeat
correcto:='s';
coge aleatorio;
aux:num[i].fra:=x;
aux:num[i].term:='y';
until(correcto='s');
gotoxy(65,3);write(i,2);
gotoxy(35,3);write(x,2,'-',y,1);
gotoxy(15,6);textcolor(15);write('*');textcolor(9);
pulsar tecla(16,6);
coger hora(h1,m1,s1,d1);
gotoxy(15,6);write(T[1][x]);
p:=length(T[1][x]);
gotoxy(p+16,6);textcolor(15);write('*');textcolor(9);
pulsar tecla(p+17,6);
gotoxy(15,6);write(*);
coger hora(h2,m2,s2,d2);
medir tiempos(h1,m1,s1,d1,h2,m2,s2,d2,h,m,s,d);
aux:med1[i].seg:=s;
aux:med1[i].cent:=d;
gotoxy(p+16,6);write(' ');
gotoxy(15,8);write(T[2][x]);
p:=length(T[2][x]);
gotoxy(p+16,8);textcolor(15);
if con then write('#');
else write('*');
textcolor(9);
pulsar tecla(p+17,8);
gotoxy(15,8);write('');
coger hora(h3,m3,s3,d3);
medir tiempos(h2,m2,s2,d2,h3,m3,s3,d3,h,m,s,d);
aux:med2[i].seg:=s;
aux:med2[i].cent:=d;
gotoxy(p+16,8);write(' '); if con then
begin
sacar texto;
resuesta;
if resp<>palabra then
aux:mal:=aux:mal+1
else
aux:biern:=aux:biern+1;
coger hora(h4,m4,s4,d4);
medir tiempos(h3,m3,s3,d3,h4,m4,s4,d4,h,m,s,d);
aux:med3[i].seg:=s;
aux:med3[i].cent:=d;
gotoxy(p+16,8);textcolor(15);write('*');textcolor(9);
pulsar tecla(p+17,8);
coger hora(h5,m5,s5,d5);
gotoxy(p+16,8);write(' '); end;
gotoxy(p+16,8);
write(T[3][x,y]);
p:=p+length(T[3][x,y]);
gotoxy(p+17,8);textcolor(15);write('*');textcolor(9);
pulsar tecla(p+18,8);
coger hora(h6,m6,s6,d6);
if con then
medir tiempos(h5,m5,s5,d5,h6,m6,s6,d6,h,m,s,d);
else
medir tiempos(h3,m3,s3,d3,h6,m6,s6,d6,h,m,s,d);
aux.med4[i].seg:=ss;
aux.med4[i].cent:=d;
end;
c1rsr;
end;

1.3.—RUTINAS SECUNDARIAS UTILIZADAS
POR LOS PROCEDIMIENTOS
PRINCIPALES.

1.3.1.—Sorteo aleatorio del número de frase
y la terminación.

El procedimiento que se presenta a conti-
uación sortea un número de frase y de ter-
minación teniendo en cuenta si la combina-
ección elegida ha sido ya presentada.

Procedure coge_aleatorio;
var x1,y1:real;
begin
x1:=int(random(24))+1;
y1:=int(random(3))+1;
x:=round(x1);
y:=round(y1);
if SUCESO[x,y]=1 then correcto:='n'
else SUCESO[x,y]:=1;
end;

1.3.2.—Definición de la tecla a utilizar para
la autoadministración de los
fragmentos de las frases.

La tecla que sirve para la auto-adminis-
tración de los fragmentos de las frases es
cursor-hacia-abajo.

Procedure pulsar_tecla(x,y:integer);
var tecla:char;
begin
repeat
  tecla:='x';
gotoxy(x,y);read(kbd,tecla);
if keypressed then
begin
  read(kbd,tecla);
  if (tecla=#27) and keypressed then read(kbd,tecla);
end;
if (tecla=#80) then
begin
  sound(2000);delay(50);nosound;
end;
until (tecla=#80);
end;

1.3.3.—Procedimiento para coger la hora del
reloj interno.

Mediante esta rutina se definen una serie
de variables que se utilizan para guardar el
tiempo. El registro AX se carga con valor de
interrupción 2C, y esto hace que el sistema
devuelva las variables CX y DX con la hora
del reloj interno.

Procedure coger_hora(var hh,mm,ss,dd:integer);
Type
  regt = record
    AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS:integer;
  end;
Var
  tregistro:regt;
begin
  with tregistro do
  begin
    AX:=2C00;
    MSDOS(tregistro);
    hh:=Hi(CX);
    mm:=Lo(CX);
    ss:=Hi(DX);
    dd:=Lo(DX);
  end;
end;

1.3.4.—Procedimiento para medir los
tiempos.

Este procedimiento es llamado también
por proceso, como el anterior, y básicamen-
te se ocupa de calcular las diferencias entre
dos tiempos (t₂-t₁) recogidos del reloj inter-
no con el fin de determinar los segundos y
las centésimas de segundo que un sujeto tarda
en leer un segmento de las frases. Los
condicionales que aparecen en esta rutina se
aplican en los casos donde esta diferencia se-
ría negativa o irreal (por ejemplo, cuando
hh2=3 y mm2=00, y hh1=2 y mm1=59, el
cálculo de una diferencia sin restricciones
establecería que hay una hora entre ambas
medidas y que existe una diferencia negati-
va respecto a los minutos).

Procedure
medir_tiempos(hh1,mm1,ss1,dd1,hh2,mm2,ss2,dd2:
integer; var h,m,s,d:integer);
begin
if mm2<mm1 then begin
   hh1:=hh1+1;
   mm2:=mm2+60;
end;
if ss2<ss1 then begin
   mm1:=mm1+1;
   ss2:=ss2+60;
end;
if dd2<dd1 then begin
   ss1:=ss1+1;
   dd2:=dd2+100;
end;
h:=hh2-hh1;m:=mm2-mm1;s:=ss2-ss1;d:=dd2-dd1;
end;

1.3.5.-Presentación de las palabras en la tarea de decisión léxica.

La rutina que se incluye en este apartado sirve para presentar los estímulos de la tarea de decisión léxica y para llevar un contador de las respuestas correctas e incorrectas de los sujetos en la misma sobre la base de la regla siguiente: si i+j es par, el estímulo es palabra, por el contrario si es impar, el estímulo es una no-palabra. Esta regla permite posteriormente saber si la respuesta es correcta o incorrecta (Véase apartado 1.2.7).

Procedure sacar_texto;
var z:integer;
begin
   gotoxy(35,12);sound(2000);delay(50);nosound; delay(50);
textcolor(15);
write(PALABRAS[x,y]);
delay(100);gotoxy(35,12);textcolor(9);write(' ');
if ((x+y) mod 2) = 0 then palabra :='s'
else palabra :='n';
end;

1.3.6.-Recogida de la respuesta en la tarea de decisión léxica.

Si se ha elegido la opción 1 (con tarea de decisión léxica) aparecerá en el momento de la decisión un mensaje como el siguiente:

   NO     ---  SI

Las teclas definidas para dar la respuesta a la tarea de decisión léxica son cursor-ia-izquierda en caso de respuesta negativa, y el cursor-hacia-la-derecha en caso de respuesta afirmativa.

Procedure respuesta;
var tecla:char;
begin
repeat
   textcolor(15);
gotoxy(30,23);write('NO     ---  SI');
textcolor(9);
tecla:= 'x';
gotoxy(40,23);read(kbd,tecla);
if keypressed then begin
   read(kbd,tecla);
   if (tecla=#27) and keypressed then read(kbd,tecla);
end;
if (tecla<>#75) and (tecla<>#77) then begin
   sound(2000);delay(50);nosound;
end;
until (tecla=#75) or (tecla=#77);
if tecla=#75 then resp:= 'n' else resp:= 's';
gotoxy(30,23);write('  ');
end;

II. IMPRESION DEL FICHERO DE RESULTADOS O REDIRECCIONAMIENTO DEL MISMO A UN FICHERO ASCII.

Los datos correspondientes a cada sujeto se van almacenando a medida que se reconocen en un único fichero (datos.dat). Los datos de todos los sujetos de un experimento pueden imprimirse llamando (fuera ya de FRA3) a un programa ejecutable que debe estar en la misma unidad donde se encuentra el fichero datos.dat. El código fuente de este programa que se ha denominado RECUENTO se presenta en el Apéndice I. Un ejemplo de una parte de un fichero de datos obtenido con FRA3 aparece en el Apéndice II.

Otra posibilidad es redireccionar la impresión del fichero de datos a un fichero ASCII mediante la utilidad PRN2FILE, que requiere asignar un nuevo nombre al fichero resultante. Esta posibilidad facilita el tratamiento de los datos con cualquier paquete de análisis estadístico.
REFERENCIAS BIBLIOGRÁFICAS


Aceptado, 29 de octubre de 1993
Program Frases;

Type
tiempo = record
    seg : integer;(* Número de segundos *)
    cent: integer;(* Número de centésimas de segundo *)
end;
numero = record
    fra : integer;(* Número de frase (1-24) *)
    term: integer;(* Número de terminacion (1,2,3) *)
end;
reg = record
    c_s : char;(* 1-Con palabras, 2-Sin palabras *)
    num : array [1..72] of numero;(* Número de frase y terminac. *)
    med1 : array [1..72] of tiempo;
    med2 : array [1..72] of tiempo;
    med3 : array [1..72] of tiempo;
    med4 : array [1..72] of tiempo;
    bien : integer;
    mal : integer;
end;

var
datos:file of reg;
aux:reg;
tecla:char;
ok:boolean;

(****************************** Fichero Inexistente ******************************)

Procedure fichero_inexistente;
var ok:boolean;
begin
    clrscr;
sound(2000);delay(60);nosound;
gotoxy(26,10);write(‘El Fichero de Datos no existe’);
gotoxy(26,11);write(‘_____________________________________________________________’);
end;

(****************************** Seguir ******************************)

Procedure seguir;
var tipo:string[14];
i:integer;
begin
    clrscr;
gotoxy(20,10);write(‘Prepare la impresora y pulse una tecla...’);
    read(kbd,tecla);
APENDICE I (Continuación)

gotoxy(20,10);write('  ');  
gotoxy(30,10);write('Imprimiendo Resultados...');
while not(eof(datos)) do 
begin
  read(datos,aux);
  writeln(lst,' '); 
  writeln(lst,' '); 
  writeln(lst,' '); 
  if aux.c_s='1' then tipo:='Con palabras'
  else tipo:='Sin palabras';
  write(lst, '  Caso : ',tipo);
  if aux.c_s='1' then  
  begin
    write(lst,' Bien : ',aux.bien:2);
    writeln(lst,' Mal : ',aux.mal:2);
  end
  else writeln(lst,' '); 
  writeln(lst,' '); 
  writeln(lst,' '); 
  writeln(lst,' Fra/Term Tiempo 1 Tiempo 2 Tiempo 3 Tiempo 4'); 
  writeln(lst,' '); 
  writeln(lst,' '); 
  for i:=1 to 72 do 
  begin
    write(lst,' ',aux.num[i].fra:2,'-','aux.num[i].term:1);
    write(lst,' ',aux.med1[i].seg:2,' : ',aux.med1[i].cent:2,'  ');
    write(lst,aux.med2[i].seg:2,' : ',aux.med2[i].cent:2,'  ');
    write(lst,aux.med3[i].seg:2,' : ',aux.med3[i].cent:2,'  ');
    writeln(lst,aux.med4[i].seg:2,' : ',aux.med4[i].cent:2);
  end;
end;
cr1sc;
gotoxy(28,10);write('Fin del listado de resultados');
end;

(**************************** Programa Principal *****************************)

Begin
assign(datos,'datos.dat');
(*$I-*)reset(datos)(*$I+*);
ok:=(ioresult=0);
if not ok then fichero_inexistente
  else seguir;
end.

Psicothema, 1994
**R. LOPEZ SANCHEZ**

**APENDICE II**

Ejemplo de un fichero de datos creado tras la aplicación de FRA3. En la cabecera aparece la opción elegida y el número de respuestas correctas e incorrectas en la tarea de decisión léxica. Debajo de la cabecera aparecen las etiquetas de cinco columnas: (1) Fra/Term: indica el número de frase y la terminación, (2) Tiempo 1: hace referencia al tiempo de lectura en segundos y centésimas del primer segmento de las frases, (3) Tiempo 2: indica el tiempo de lectura para el segundo fragmento, (4) Tiempo 3: sería el tiempo que el sujeto tarda en tomar la decisión léxica (este tiempo sería cero si la opción elegida fuera la 2), por último, (5) Tiempo 4: sería el tiempo de lectura del tercer segmento de las frases (terminación).

<table>
<thead>
<tr>
<th>Caso:</th>
<th>Con palabras</th>
<th>Bien : 54</th>
<th>Mal : 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fra/Term</td>
<td>Tiempo 1</td>
<td>Tiempo 2</td>
<td>Tiempo 3</td>
</tr>
<tr>
<td>3-2</td>
<td>2 : 36</td>
<td>1 : 48</td>
<td>2 : 47</td>
</tr>
<tr>
<td>22-3</td>
<td>1 : 65</td>
<td>1 : 98</td>
<td>1 : 54</td>
</tr>
<tr>
<td>2-4</td>
<td>2 : 85</td>
<td>1 : 32</td>
<td>2 : 3</td>
</tr>
<tr>
<td>21-3</td>
<td>2 : 37</td>
<td>1 : 31</td>
<td>1 : 65</td>
</tr>
<tr>
<td>12-2</td>
<td>1 : 15</td>
<td>1 : 26</td>
<td>1 : 27</td>
</tr>
<tr>
<td>16-1</td>
<td>1 : 82</td>
<td>1 : 26</td>
<td>1 : 21</td>
</tr>
<tr>
<td>20-1</td>
<td>2 : 31</td>
<td>1 : 32</td>
<td>2 : 63</td>
</tr>
<tr>
<td>15-1</td>
<td>2 : 42</td>
<td>1 : 65</td>
<td>2 : 47</td>
</tr>
<tr>
<td>19-3</td>
<td>1 : 86</td>
<td>1 : 65</td>
<td>1 : 10</td>
</tr>
<tr>
<td>9-3</td>
<td>1 : 48</td>
<td>1 : 82</td>
<td>1 : 54</td>
</tr>
<tr>
<td>2-3</td>
<td>2 : 14</td>
<td>2 : 64</td>
<td>1 : 42</td>
</tr>
<tr>
<td>23-1</td>
<td>1 : 65</td>
<td>1 : 10</td>
<td>1 : 92</td>
</tr>
<tr>
<td>21-1</td>
<td>1 : 60</td>
<td>1 : 75</td>
<td>1 : 71</td>
</tr>
<tr>
<td>22-2</td>
<td>1 : 71</td>
<td>1 : 59</td>
<td>1 : 43</td>
</tr>
<tr>
<td>18-1</td>
<td>1 : 87</td>
<td>2 : 14</td>
<td>1 : 49</td>
</tr>
<tr>
<td>18-2</td>
<td>2 : 4</td>
<td>1 : 86</td>
<td>1 : 5</td>
</tr>
<tr>
<td>7-3</td>
<td>2 : 20</td>
<td>1 : 48</td>
<td>1 : 21</td>
</tr>
<tr>
<td>18-3</td>
<td>1 : 92</td>
<td>1 : 48</td>
<td>2 : 31</td>
</tr>
<tr>
<td>14-2</td>
<td>1 : 75</td>
<td>1 : 10</td>
<td>1 : 10</td>
</tr>
<tr>
<td>13-2</td>
<td>2 : 47</td>
<td>1 : 76</td>
<td>1 : 15</td>
</tr>
<tr>
<td>13-3</td>
<td>1 : 75</td>
<td>1 : 32</td>
<td>1 : 5</td>
</tr>
<tr>
<td>24-2</td>
<td>1 : 81</td>
<td>1 : 10</td>
<td>1 : 21</td>
</tr>
<tr>
<td>1-1</td>
<td>1 : 38</td>
<td>1 : 15</td>
<td>0 : 93</td>
</tr>
<tr>
<td>4-1</td>
<td>1 : 65</td>
<td>1 : 38</td>
<td>1 : 48</td>
</tr>
<tr>
<td>17-1</td>
<td>1 : 21</td>
<td>1 : 10</td>
<td>1 : 48</td>
</tr>
<tr>
<td>23-2</td>
<td>1 : 59</td>
<td>1 : 43</td>
<td>1 : 32</td>
</tr>
<tr>
<td>24-1</td>
<td>1 : 48</td>
<td>1 : 15</td>
<td>1 : 60</td>
</tr>
<tr>
<td>12-1</td>
<td>1 : 49</td>
<td>1 : 42</td>
<td>1 : 98</td>
</tr>
<tr>
<td>7-1</td>
<td>1 : 21</td>
<td>1 : 31</td>
<td>1 : 21</td>
</tr>
<tr>
<td>19-2</td>
<td>1 : 70</td>
<td>1 : 43</td>
<td>1 : 76</td>
</tr>
<tr>
<td>9-1</td>
<td>1 : 32</td>
<td>1 : 15</td>
<td>0 : 99</td>
</tr>
<tr>
<td>11-3</td>
<td>1 : 37</td>
<td>1 : 21</td>
<td>1 : 59</td>
</tr>
<tr>
<td>10-1</td>
<td>1 : 43</td>
<td>1 : 43</td>
<td>1 : 32</td>
</tr>
<tr>
<td>10-2</td>
<td>1 : 21</td>
<td>1 : 32</td>
<td>1 : 26</td>
</tr>
<tr>
<td>2-2</td>
<td>1 : 4</td>
<td>1 : 43</td>
<td>1 : 32</td>
</tr>
<tr>
<td>15-3</td>
<td>1 : 15</td>
<td>1 : 21</td>
<td>1 : 32</td>
</tr>
</tbody>
</table>