Programación 1

Problemas 1

Notación Backus-Naur

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Notación de Backus-Naur

- Notación BNF (*Backus-Naur form*)
  - Definición de reglas sintácticas para definir lenguajes
    - En particular, del lenguaje C++
  - Descripción de la organización de estructuras de datos secuenciales (ficheros)

Notación de Backus-Naur

- **Metasímbolos** utilizados:
  - Definición de una regla
    <nombre_regla> ::= expresión
  - Sustitución de la expresión
    <nombre_regla>
  - Literal
    “Prog1f”
  - Alternativa
    expresión1 | expresión2
  - Agrupación sin repetición
    (expresión)
  - Agrupación con opcionalidad (cero o una veces)
    [expresión]
  - Agrupación con repetición (cero, una o más veces)
    { expresión }

Notación Backus-Naur

: : =	Definición de regla sintáctica
< >	Delimitadores de nombre de regla sintáctica
“ ”	Delimitadores de carácter o secuencia de caracteres literal
|	Separador de alternativas
( )	Agrupador sin repetición
{ }	Agrupador con repetición (0, 1 o más veces)
[ ]	Agrupador opcional (0 o 1 vez)

Ejemplos «lingüísticos»

```
<oración> ::= [<sujeto>] <predicado>
<sujeto> ::= <sintagma_nominal>
<predicado> ::=
     (<verbo_copulativo> <atributo>)
    |(<verbo_predicativo>
      [<complemento_directo>]
      [<complemento_indirecto>]
      {<complemento_circunstancial>})
```

Ejemplo

```
<letra> ::= <vocal> | <consonante>
<vocal> ::= “a” | “e” | “i” | “o”
        | “u” | “á” | “é” | “í” | “ó”
        | “ú” | “ü”
<consonante> ::= “b” | “c” | “d”
        | “f” | “g” | “h” | “j” | “k”
        | “l” | “m” | “n” | “ñ” | “p”
        | “q” | “r” | “s” | “t” | “v”
        | “w” | “x” | “y” | “z”
```

Ejemplo

<palabra> ::=

Ejemplo

```
<palabra> ::= <letra> {<letra>}
```

Problema 1

- Utilizando las reglas `<vocal>` y `<consonante>` del ejemplo anterior:
- **Palabras que empiezan por consonante y tienen vocales y consonantes alternadas**
  - g, de, pan, caso, hogar, coraza, mirador, zaragozano, …

Problema 2

```
<mayúscula> ::= “A” | “B” | “C” | “D”
      | “E” | “F” | “G” | “H” | “I” | “J”
      | “K” | “L” | “M” | “N” | “O” | “P”
      | “Q” | “R” | “S” | “T” | “U” | “V”
      | “W” | “X” | “Y” | “Z” | “Ñ” | “Á”
      | “É” | “Í” | “Ó” | “Ú” | “Ü”
<minúscula> ::= “a” | “b” | “c” | “d”
      | “e” | “f” | “g” | “h” | “i” | “j”
      | “k” | “l” | “m” | “n” | “o” | “p”
      | “q” | “r” | “s” | “t” | “u” | “v”
      | “w” | “x” | “y” | “z” | “ñ” | “á”
      | “é” | “í” | “ó” | “ú” | “ü”
```

Problema 2

- Palabras con secuencia correcta de mayúsculas y minúsculas

Problema 2. Solución

```
<palabra> ::=
      (<minúscula> {<minúscula>})
    | (<mayúscula> {<minúscula>})
    | (<mayúscula> {<mayúscula>})
```

Problema 3. Sintaxis en notación NBF para un número entero positivo

```
<dígito> ::= “0” | “1” | “2” | “3”
        | “4” | “5” | “6” | “7” | “8”
        | “9”

<natural>
        ::= <dígito> {<dígito>}
```

Problema 3. Sintaxis en notación NBF para un número entero positivo

```
<dígitoNoNulo> ::= “1” | “2” | “3”
        | “4” | “5” | “6” | “7” | “8”
        | “9”
<dígito> ::= “0” | <dígitoNoNulo>
<natural> ::= “0”
        | (<dígitoNoNulo> {<dígito>})
```

Problema 4. Secuencias binarias

```
<bit> ::= “0” | “1”

<secuencia_binaria>
    ::= <bit> {<bit>}
```

Problema 5. Identificadores

```
<identificador> ::=
   ( <letra>|“_” ) { <letra>|<dígito>|“_” }
<letra> ::= <mayúscula> | <minúscula>
<mayúscula> ::= “A”|“B”|“C”|“D”|“E”|“F”|“G”|“H”
  |“I”|“J”|“K”|“L”|“M”|“N”|“O”|“P”|“Q”|“R”|“S”
  |“T”|“U”|“V”|“W”|“X”|“Y”|“Z”
<minúscula> ::= “a”|“b”|“c”|“d”|“e”|“f”|“g”|“h”
  |“i”|“j”|“k”|“l”|“m”|“n”|“o”|“p”|“q”|“r”|“s”
  |“t”|“u”|“v”|“w”|“x”|“y”|“z”
<dígito> ::= “0”|“1”|“2”|“3”|“4”|“5”|“6”|“7”
  |“8”|“9”
```

Problema 5. Identificadores

```
<identificador> ::=
   ( <letra>| _ ) { <letra> | <dígito> | _ }
<letra> ::= <mayúscula> | <minúscula>
<mayúscula> ::=  A | B | C | D | E | F | G | H
  | I | J | K | L | M | N | O | P | Q | R | S
  | T | U | V | W | X | Y | Z
<minúscula> ::=  a | b | c | d | e | f | g | h
  | i | j | k | l | m | n | o | p | q | r | s
  | t | u | v | w | x | y | z
<dígito> ::=  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
  | 8 | 9
```

Problema 5. Identificadores

(¿válidos, no válidos, aconsejables?)

C++	o123
java	mannana
UMO164G	mañana
spider-man	interés
pRINCIPIO	cœur
error!	tasa de cambio
77E2	p'adelante
String	hinundaciones
begin	cero07

Problema 5. Identificadores

(¿válidos, no válidos, aconsejables?)

```
dato_leido
primer_dato_escrito_en_el_fichero
primerDato
dinero$
_aux
__aux
_Aux
tabla_temperaturas_____auxiliar
X_
o_o
```

Problema 6. Estructuras de datos
Ficheros con formato SubRip (.srt)

```
...

104
00:08:02,997 --> 00:08:05,563
Sí, bueno, creía que la Mano del Rey...

105
00:08:05,663 --> 00:08:07,541
...era bienvenido en las
reuniones del Pequeño Consejo.

106
00:08:07,575 --> 00:08:09,577
Nuestro padre es la Mano del Rey.

107
00:08:09,611 --> 00:08:12,647
Sí, pero en su ausencia...

...
```

Problema 6. Estructuras de datos
Ficheros con formato SubRip (.srt)

```
<ficheroSubRip> ::= {<subtítulo>}
<subtítulo> ::= <número> finLínea
    <tiempo> " --> " <tiempo> finLínea
    {<texto> finLínea}
    finLínea
<número> ::= <dígito> {<dígito>}
<tiempo> ::= <dígito> <dígito> ":"
    <dígito> <dígito> ":" <dígito> <dígito> ","
    <dígito> <dígito> <dígito>
<texto> ::= <carácter> {<carácter>}
<carácter> ::= carácter distinto a finLínea
```