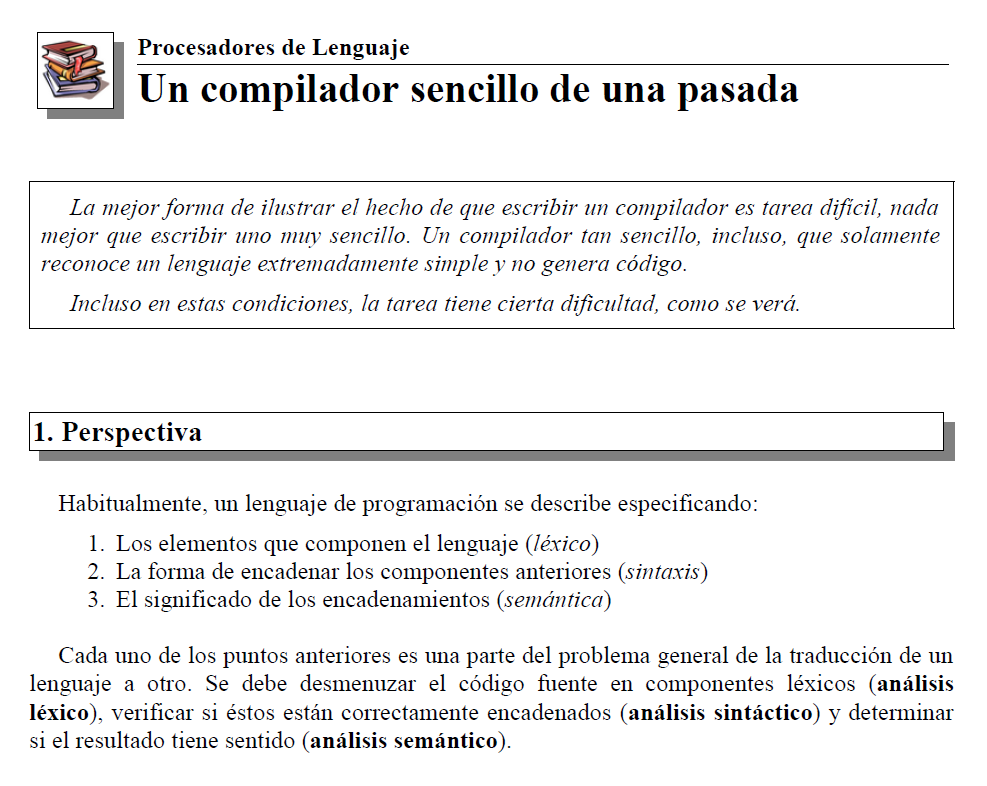
## Unidad II: Idiomas y su especificación formal. Forma Backus-Naur.



**Notación de Backus-Naur**

[**http://es.wikipedia.org/wiki/Notaci%C3%B3n\_de\_Backus-Naur**](http://es.wikipedia.org/wiki/Notaci%C3%B3n_de_Backus-Naur)

La notación de Backus-Naur, también conocida por sus denominaciones inglesas Backus-Naur form (BNF), Backus-Naur formalismo Backus normal form, es un [metalenguaje](http://es.wikipedia.org/wiki/Metalenguaje) usado para expresar [gramáticas libres de contexto](http://es.wikipedia.org/wiki/Gram%C3%A1tica_libre_de_contexto): es decir, una manera formal de describir [lenguajes formales](http://es.wikipedia.org/wiki/Lenguaje_formal).

El BNF se utiliza extensamente como notación para las [gramáticas](http://es.wikipedia.org/wiki/Gram%C3%A1tica) de los [lenguajes de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de la computadora, de los [sistemas de comando](http://es.wikipedia.org/w/index.php?title=Sistema_de_comando&action=edit&redlink=1) y de los [protocolos de comunicación](http://es.wikipedia.org/wiki/Protocolo_de_comunicaci%C3%B3n), así como una notación para representar partes de las gramáticas de la [lengua natural](http://es.wikipedia.org/wiki/Lengua_natural) (por ejemplo, el metro en la poesía de Venpa). La mayoría de los libros de textos para la teoría o la semántica del lenguaje de programación documentan el lenguaje de programación en BNF.

Algunas variantes, tales como la Augmented Backus-Naur Form (ABNF) y la Extended Backus–Naur Form (EBNF), tienen su propia documentación.

La idea de transcribir la estructura del lenguaje con reglas de reescritura se remontan cuando menos al trabajo del gramático indio [Panini](http://es.wikipedia.org/wiki/Panini_%28hindu%29) (hacia el [460 a. C.](http://es.wikipedia.org/wiki/A%C3%B1os_460_a._C.)), que la utilizó en su descripción de la estructura de palabras del [idioma sánscrito](http://es.wikipedia.org/wiki/Idioma_s%C3%A1nscrito) (algunos incluso han sugerido renombrar BNF a Forma Panini-Backus). Lingüïstas estadounidenses como Leonard Bloomfield y Zellig Harris llevaron esta idea un paso más adelante al tratar de formalizar el lenguaje y su estudio en términos de definiciones formales y procedimientos (1920-1960).

[Noam Chomsky](http://es.wikipedia.org/wiki/Noam_Chomsky), maestro de lingüística de alumnos de teoría de la información del [MIT](http://es.wikipedia.org/wiki/MIT), combinó la lingüística y las matemáticas, tomando esencialmente el formalismo de Axel Thue como la base de su descripción de la sintaxis del lenguaje natural. También introdujo una clara distinción entre reglas generativas (de la gramática libre de contexto) y reglas transformativas (1956).

[John Backus](http://es.wikipedia.org/wiki/John_Backus), un diseñador de lenguajes de programación de [IBM](http://es.wikipedia.org/wiki/IBM), adoptó las reglas generativas de Chomsky para describir la sintaxis del nuevo lenguaje de programación IAL, conocido en la actualidad como [ALGOL 58](http://es.wikipedia.org/w/index.php?title=ALGOL_58&action=edit&redlink=1) (1959), presentando en el primer Congreso de Computación Mundial ([World Computer Congress](http://es.wikipedia.org/w/index.php?title=World_Computer_Congress&action=edit&redlink=1" \o "World Computer Congress (aún no redactado))) el artículo "The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference".

Peter Naur, en su reporte sobre ALGOL 60 de 1963, identificó la notación de Backus como la Forma Normal de Backus (Backus Normal Form), y la simplificó para usar un conjunto de símbolos menor, pero a sugerencia de [Donald Knuth](http://es.wikipedia.org/wiki/Donald_Knuth), su apellido fue agregado en reconocimiento a su contribución, reemplazando la palabra "Normal" por Naur, dado que no se trata de una forma normal en ningún sentido, a diferencia, por ejemplo de la Forma Normal de Chomsky.

Una especificación de BNF es un sistema de reglas de derivación, escrito como:

<simbolo> ::= <expresión con símbolos>

donde <[símbolo](http://es.wikipedia.org/wiki/S%C3%ADmbolo)> es un [*no terminal*](http://es.wikipedia.org/w/index.php?title=No_terminal&action=edit&redlink=1), y la [expresión](http://es.wikipedia.org/w/index.php?title=Expresi%C3%B3n_%28matem%C3%A1ticas%29&action=edit&redlink=1) consiste en secuencias de símbolos o secuencias separadas por la [barra vertical](http://es.wikipedia.org/wiki/Barra_vertical), '|', indicando una [opción](http://es.wikipedia.org/wiki/Opci%C3%B3n), el conjunto es una posible [substitución](http://es.wikipedia.org/wiki/Substituci%C3%B3n) para el símbolo a la izquierda. Los símbolos que nunca aparecen en un lado izquierdo son [*terminales*](http://es.wikipedia.org/w/index.php?title=S%C3%ADmbolo_terminal&action=edit&redlink=1).

## Ejemplo

Como ejemplo, considere este BNF para una [dirección postal](http://es.wikipedia.org/w/index.php?title=Direcci%C3%B3n_%28geograf%C3%ADca%29&action=edit&redlink=1)

<dirección postal> ::= <nombre\_destinatario> <dirección> <apartado postal>

<personal> ::= <primer nombre> | <inicial> "."

<nombre\_destinatario> ::= <personal> <apellido> [<trato>] <EOL>

| <personal> <nombre\_destinatario>

<nombre\_de\_calle> ::= <personal> <apellido> [<trato>]

| <personal> <nombre\_de\_calle>

<dirección> ::= [<Acronimo>] <nombre de la calle> <número de la casa>

<EOL>

<Acronimo> ::= “Avda.” | “c/” | “Plza” | “Pº”

<nombre de la calle> ::= <calle> | <calle> <nombre\_de\_calle>

<calle> ::= <primer nombre>

<apartado postal> ::= <ciudad> "," <código postal> <EOL> |

<código postal> <ciudad> <EOL>

<ciudad> ::= <primer nombre> | <primer nombre> <ciudad>

**Nota: Si añado esta producción permitiría tener direcciones con número de casa “sn” o “s/n” (sin número)**

<número de la casa> ::= (1-9)(0-9)\* | “sn” | “s/n”

Ejemplo: Según este par de reglas vamos a deducir los nombres válidos y no válidos para incluir en direcciones postales.

<personal> ::= <primer nombre> | <inicial> "."

<nombre> ::= <personal> <apellido> [<trato>] <EOL>

| <personal> <nombre>

D. Diego G.

Dispararía la regla <nombre> ::= <personal> <nombre>

Que cambiaría <personal> por <inicial> "." (D.) seguida de otro disparo recursivo de <nombre> que bien podría ser nuevamente <nombre> ::= <personal> <nombre> y ahora <personal> por <primer nombre> que sería “Diego” pero quedaría por volver a disparar <nombre> de manera recursiva que no es capaz de dar solución para G. Cualquier otra combinación posible (con apellido) tampoco lo resuelve, por lo tanto el nombre D. Diego G. es un nombre NO válido para una dirección postal.

D. Diego Genzor

Dispararía la regla <nombre> ::= <personal> <nombre>

Que cambiaría <personal> por <inicial> "." (D.) seguida de otro disparo recursivo de <nombre> que en este caso podría ser <nombre> ::= <personal> <apellido>. Ahora <personal> lo cambio por <primer nombre> que sería “Diego” y quedaría <apellido> que no es capaz de dar otro que Genzor. Seguido viene ya el salto de línea <EOL> por lo que resuelve de manera satisfactoria el nombre, y por lo tanto el nombre D. Diego Genzor ES un nombre VALIDO para una dirección postal.

Diego Genzor

D.

<nombre> ::= <personal> <nombre>

Segundo disparo de regla <nombre> (recursivo)

Diego

Genzor

<nombre> ::= <personal> <nombre>??? 🡪 Noooo !!! A priori se ve que nombre obligará a tener dos elementos por lo que como sólo tengo un valor se seguro que va a fallar . . . por lo tanto elijo la otra opción que tiene como opcional el trato (por lo tanto no lo pondré) y el EOL que es el salto de linea.

Diego

Genzor

<nombre> ::= <personal> <apellido> [<trato>] <EOL>

Y finalmente quedará que “D. Diego Genzor” es un nombre válido compuesto por:

<inicial> "." <primer nombre> <apellido> <EOL>

Esto se traduce a español como:

* Una dirección postal consiste en un nombre, seguido por una dirección, seguida por un apartado postal.
* Una parte "personal" consiste en un nombre o una inicial seguido(a) por un punto.
* Un nombre consiste de: una parte personal seguida por un apellido seguido opcionalmente por una jerarquía o el trato que se la da a la persona (Jr., Sr., o número dinástico) y un salto de línea (end-of-line), o bien una parte personal seguida por un nombre (esta regla ilustra el uso de la repetición en BNFs, cubriendo el caso de la gente que utiliza múltiples nombres y los nombres medios o las iniciales).
* Una dirección consiste de una especificación opcional del departamento, seguido de un número de casa, seguido por el nombre de la calle, seguido por un salto de línea (end-of-line).
* Un apartado postal consiste de una ciudad, seguida por una coma, seguida por un código del estado (recuerde que es un ejemplo que ocurre en EE.UU.), seguido por un código postal y este seguido por un salto de línea (end-of-line).

Observe que muchas cosas (tales como el formato de una parte personal, de una especificación del apartamento, o código postal) están dejadas sin especificar aquí. Si es necesario, pueden ser descritas usando reglas adicionales de BNF, o dejadas como [abstracción](http://es.wikipedia.org/wiki/Abstracci%C3%B3n_%28inform%C3%A1tica%29) si es inaplicable para el propósito actual.

## Otros ejemplos

Bastante interesante, el ejemplo de cómo interpretar una operación de asignación de una variable con una constante de tipo entero con un final de línea que sea un “;” y tras él pueda haber signos de tabulación y/o espacios en blanco antes del salto de línea, sería como sigue:

<asignación> ::= <identificador> <operador\_asignacion> <constante\_entera><fin>[<no\_visibles>]<EOL>

<identificador> ::= ((a-z)|(A-Z))( (a-z)|(A-Z)|(0-9)|”\_”)\*

<operador\_asignacion> ::= “=”

<constante\_entera> ::= ((1-9)(0-9)\*)|0

<fin> ::= “;”

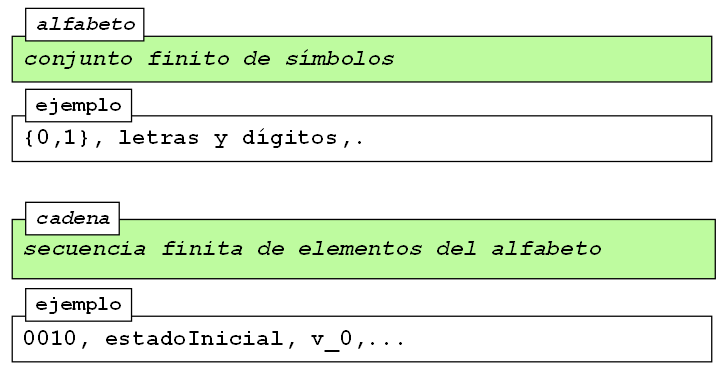
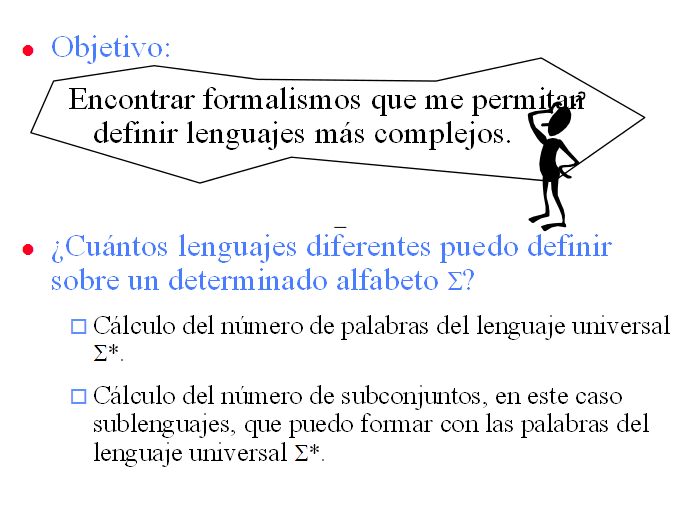
<no\_visibles> ::= “ “[<no\_visibles>] | <TAB>[<no\_visibles>]

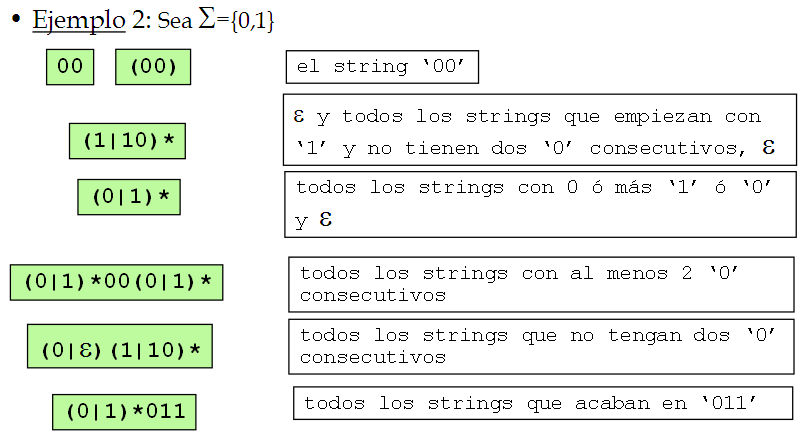
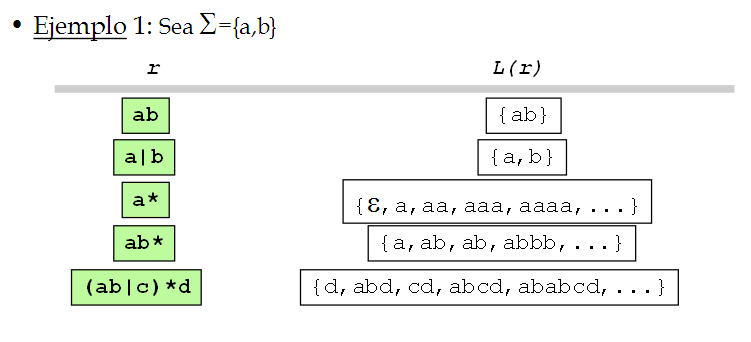
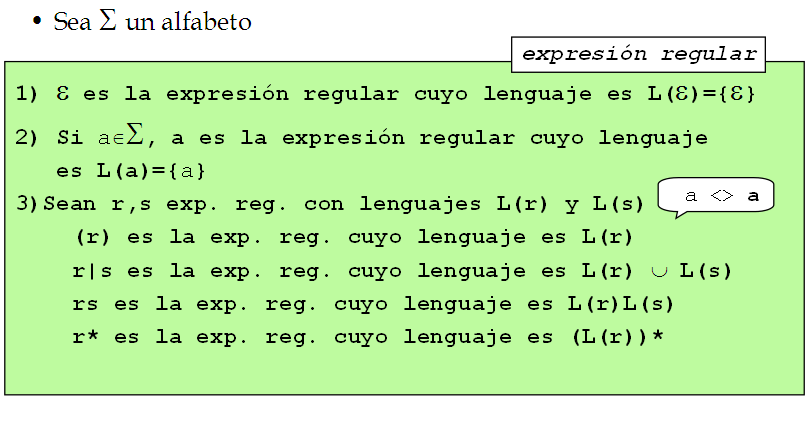
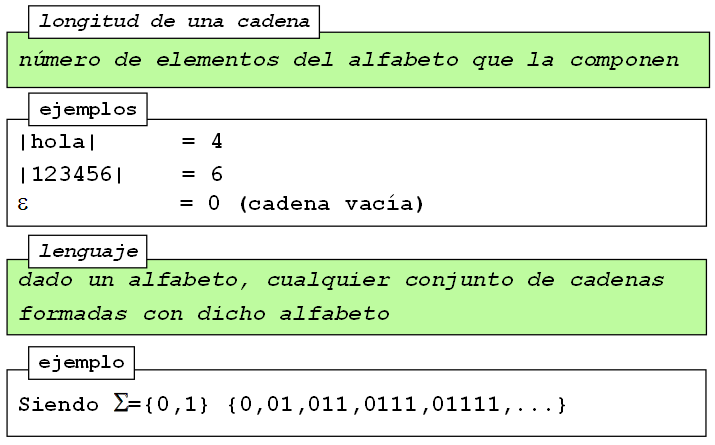
<no\_visibles> ::= “ “|<TAB>|“ “<no\_visibles> | <TAB><no\_visibles> /\*Equivalente al anterior\*/

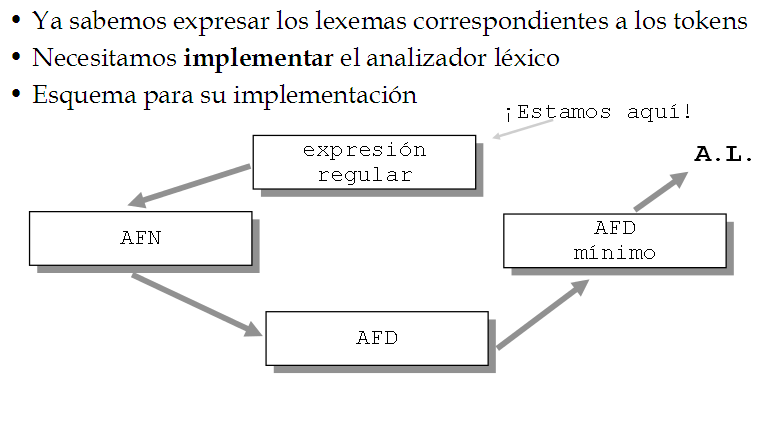
<no\_visibles> ::= (“ “| <TAB>)(“ “| <TAB>)\* /\* Este es equivalente al anterior pero sin recursividad\*/

## Definición de lenguajes

[Manual de SQL de MySql (web de Oracle)](http://docs.oracle.com/cd/E17952_01/refman-5.5-en/sql-syntax.html)







Así las expresiones regulares nos sirven para definir las reglas finales en expresiones BNF. Por ejemplo el caso de la pg 3 donde reconocíamos nombres y apellidos:

Diego

Genzor

<nombre> ::= <personal> <apellido> [<trato>] <EOL>

Y finalmente quedará que “D. Diego Genzor” es un nombre válido compuesto por:

<inicial> "." <primer nombre> <apellido> <EOL>

Ahora podríamos añadir las reglas:

<primer nombre**>** ::= [A-Z][a-z]\*

<apellido>::= [A-Z][a-z]\*

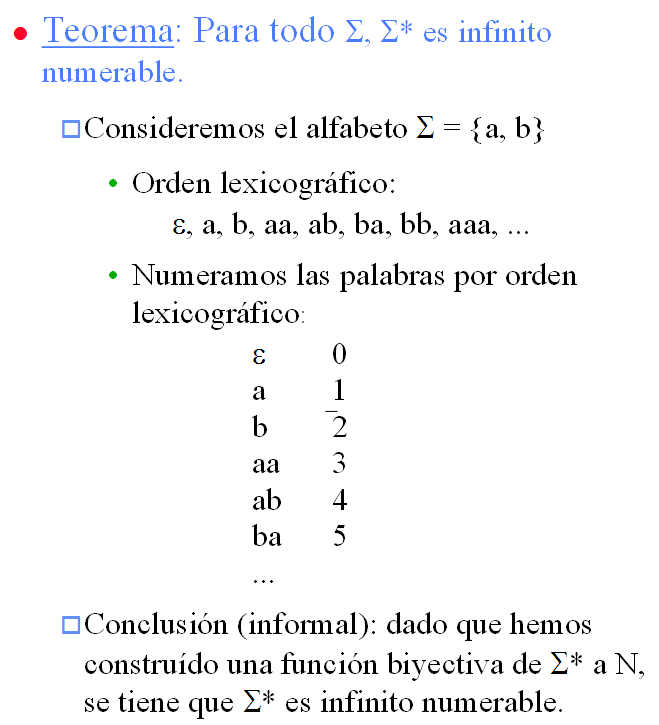
<inicial>::= [A-Z]

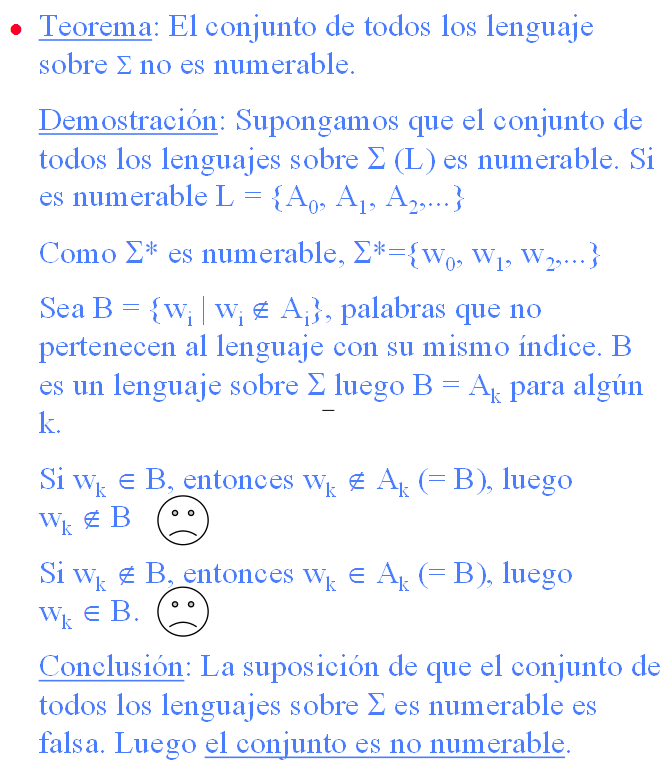
Ahora podemos poner en modo europeo los tratamientos de los nombres y modificar esta regla añadiendo alguna mas:

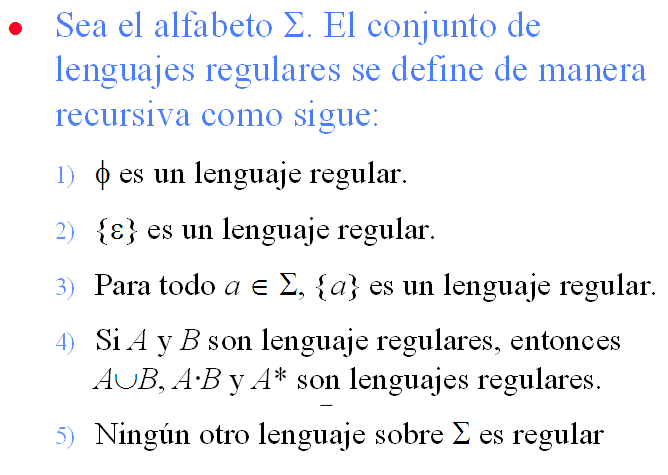
<nombre> ::= [<trato>] <personal> <apellido> <EOL> | <personal> <nombre>

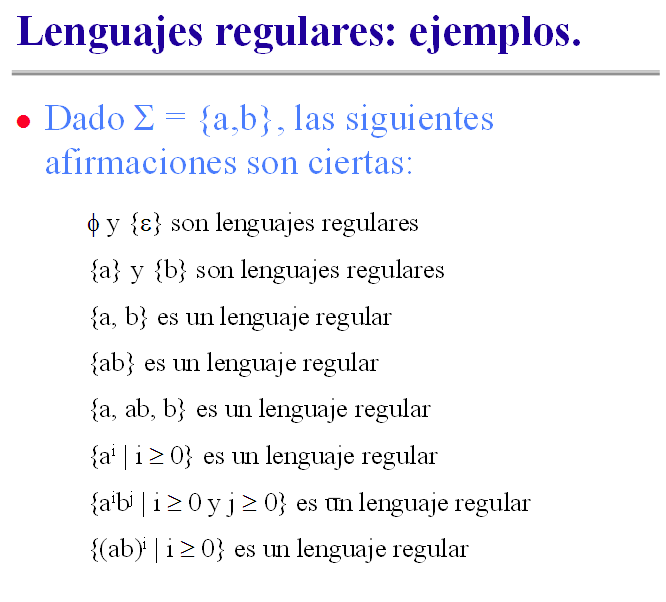
<trato>::= “Sr.” | “Sra.” | “D.” | “Dña.” | “Excmo.”

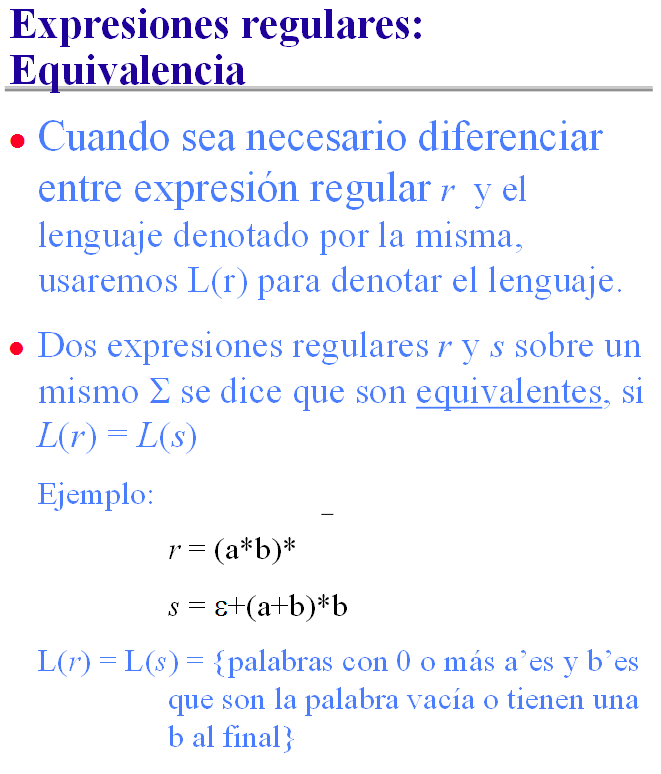
Es decir, no por haber visto ahora las expresiones regulares intentemos definir todo así, si hay conjuntos de elementos finitos y no muy extensos se pueden poner como enumerados.











|

|

