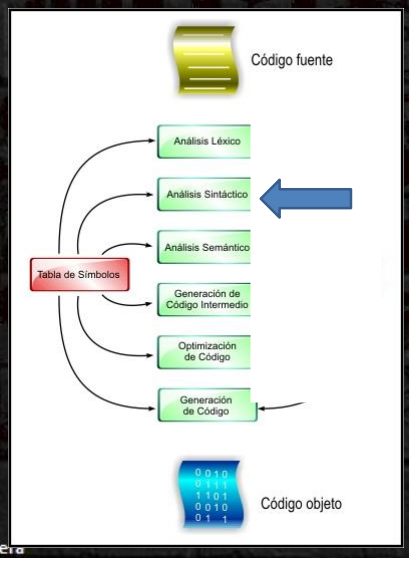
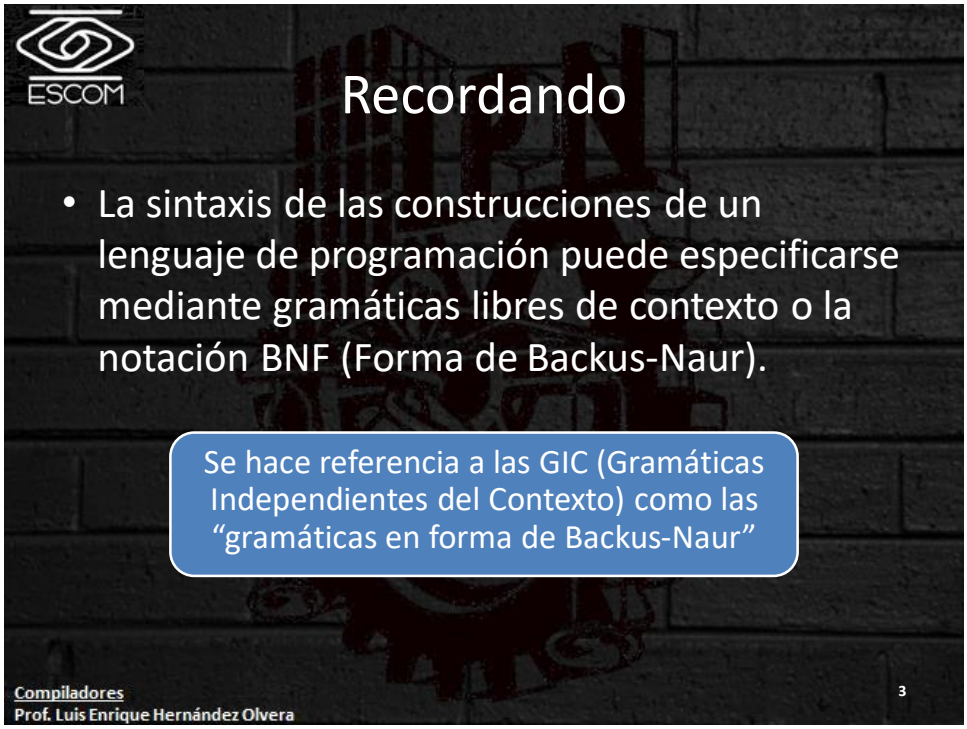
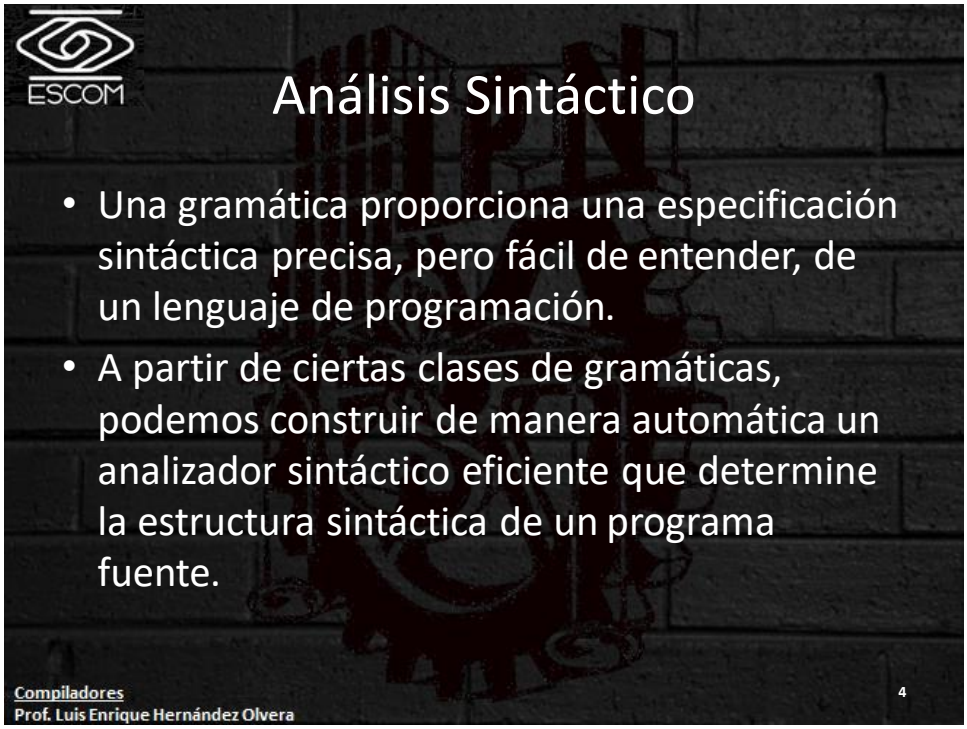
Analizador Sintáctico

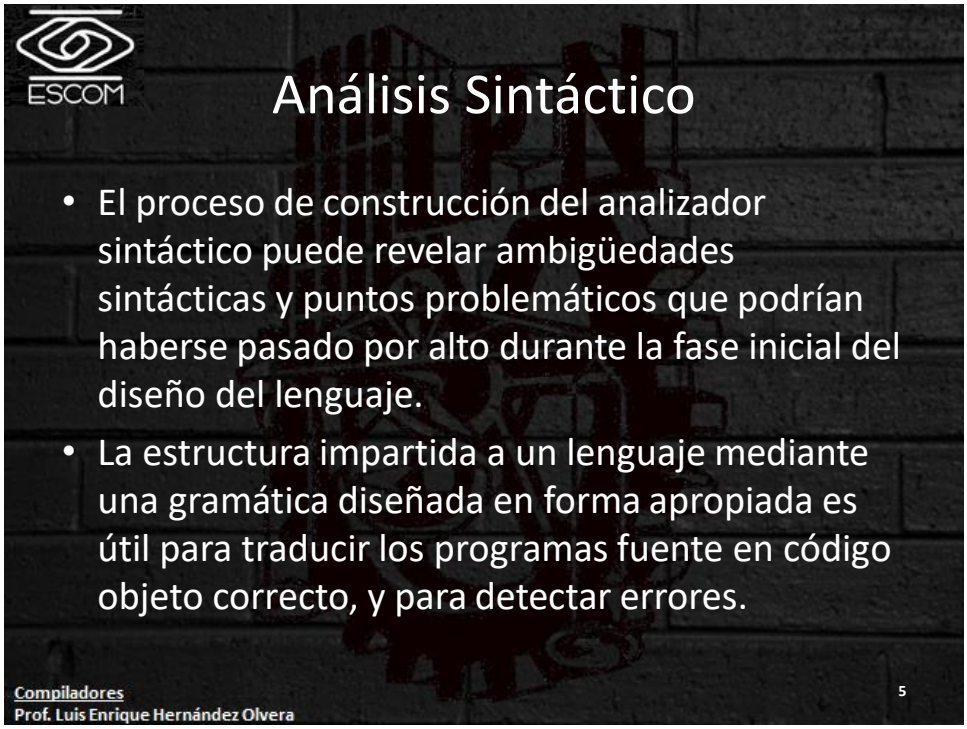


Recordando

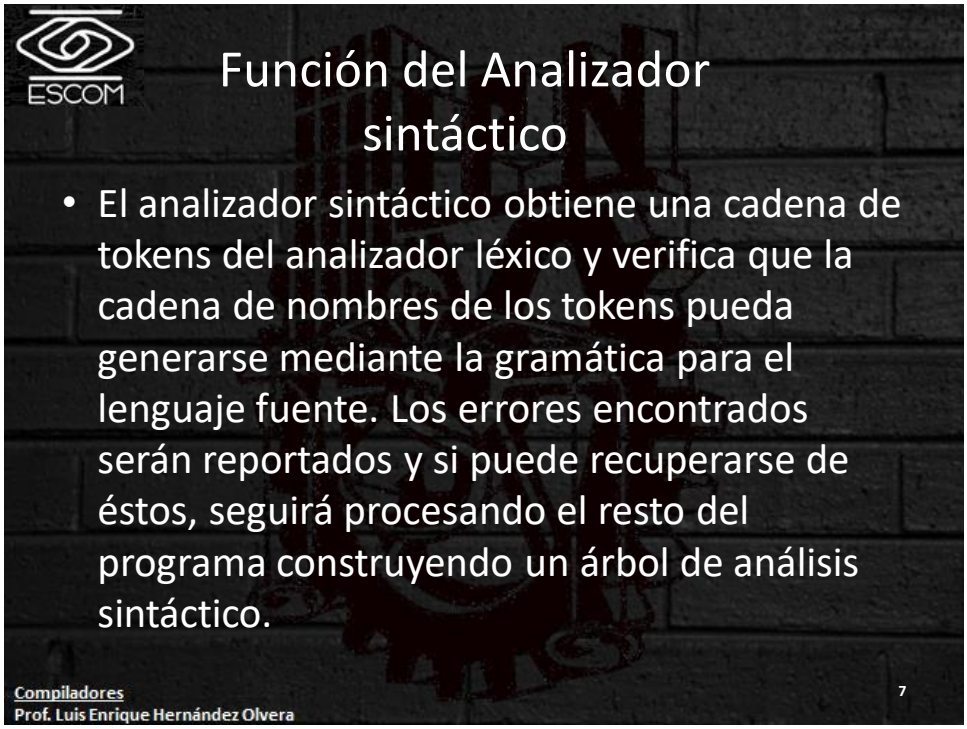
- La **gramática** es el **estudio de las reglas y principios** que regulan el uso de las lenguas y la organización de las palabras dentro de una oración. También se denomina así al "**conjunto de reglas y principios que gobiernan el uso de un lenguaje**" así, cada lenguaje tiene su propia gramática.

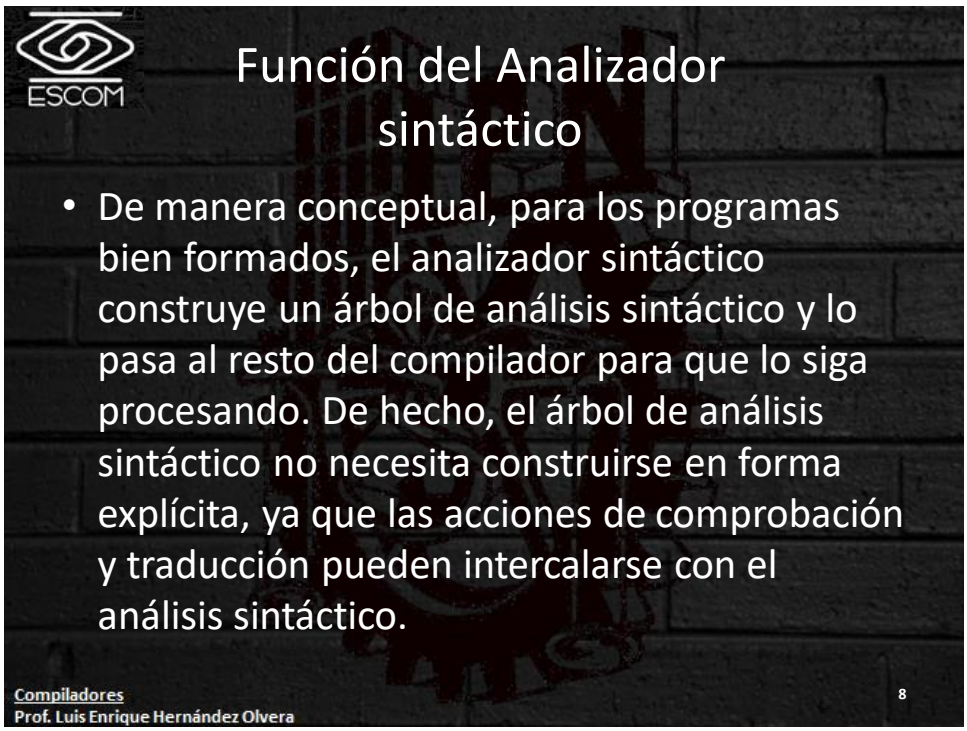

ESCOM
Recordando
• La sintaxis de las construcciones de un lenguaje de programación puede especificarse mediante gramáticas libres de contexto o la notación BNF (Forma de Backus-Naur).
Se hace referencia a las GIC (Gramáticas Independientes del Contexto) como las "gramáticas en forma de Backus-Naur"
Compiladores
Prof. Luis Enrique Hernández Olvera
3


ESCOM
Análisis Sintáctico
• Una gramática proporciona una especificación sintáctica precisa, pero fácil de entender, de un lenguaje de programación.
• A partir de ciertas clases de gramáticas, podemos construir de manera automática un analizador sintáctico eficiente que determine la estructura sintáctica de un programa fuente.
Compiladores
Prof. Luis Enrique Hernández Olvera
4

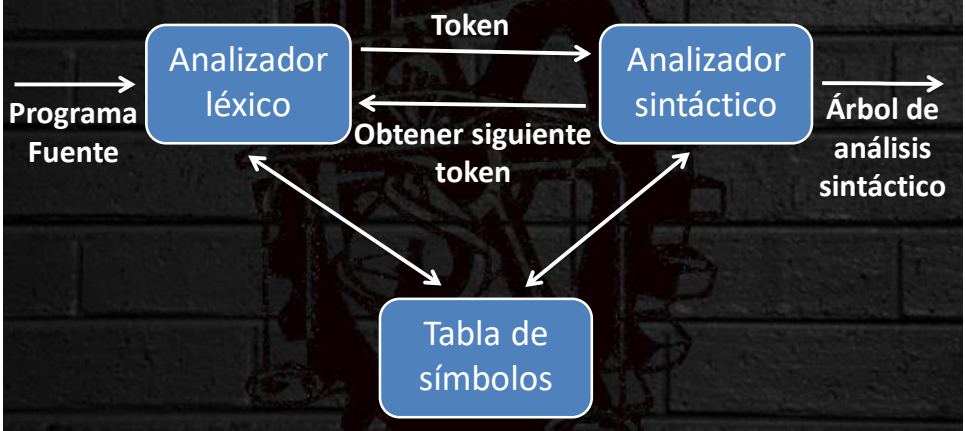

ESCOM
Análisis Sintáctico
• El proceso de construcción del analizador sintáctico puede revelar ambigüedades sintácticas y puntos problemáticos que podrían haberse pasado por alto durante la fase inicial del diseño del lenguaje.
• La estructura impartida a un lenguaje mediante una gramática diseñada en forma apropiada es útil para traducir los programas fuente en código objeto correcto, y para detectar errores.
Compiladores
Prof. Luis Enrique Hernández Olvera
5


ESCOM
Análisis Sintáctico
• Una gramática permite que un lenguaje evolucione o se desarrolle en forma iterativa, agregando nuevas construcciones para realizar nuevas tareas. Estas nuevas construcciones pueden integrarse con más facilidad en una implementación que siga la estructura gramatical del lenguaje.
Compiladores
Prof. Luis Enrique Hernández Olvera
6


ESCOM
Función del Analizador sintáctico
• El analizador sintáctico obtiene una cadena de tokens del analizador léxico y verifica que la cadena de nombres de los tokens pueda generarse mediante la gramática para el lenguaje fuente. Los errores encontrados serán reportados y si puede recuperarse de éstos, seguirá procesando el resto del programa construyendo un árbol de análisis sintáctico.
Compiladores
Prof. Luis Enrique Hernández Olvera
7

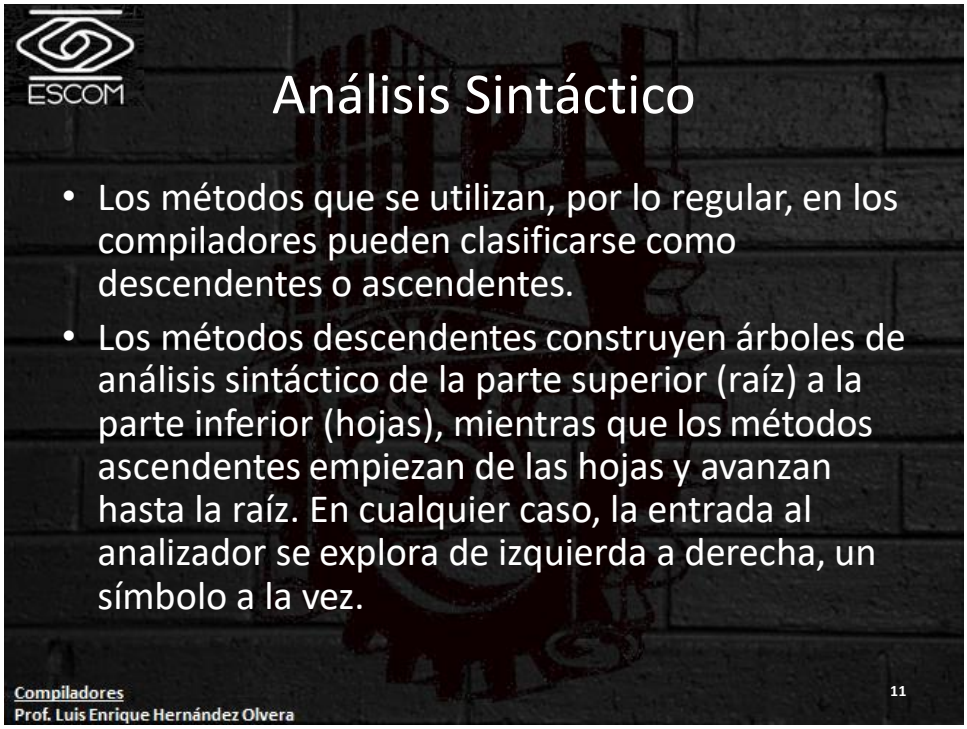

ESCOM
Función del Analizador sintáctico
• De manera conceptual, para los programas bien formados, el analizador sintáctico construye un árbol de análisis sintáctico y lo pasa al resto del compilador para que lo siga procesando. De hecho, el árbol de análisis sintáctico no necesita construirse en forma explícita, ya que las acciones de comprobación y traducción pueden intercalarse con el análisis sintáctico.
Compiladores
Prof. Luis Enrique Hernández Olvera
8

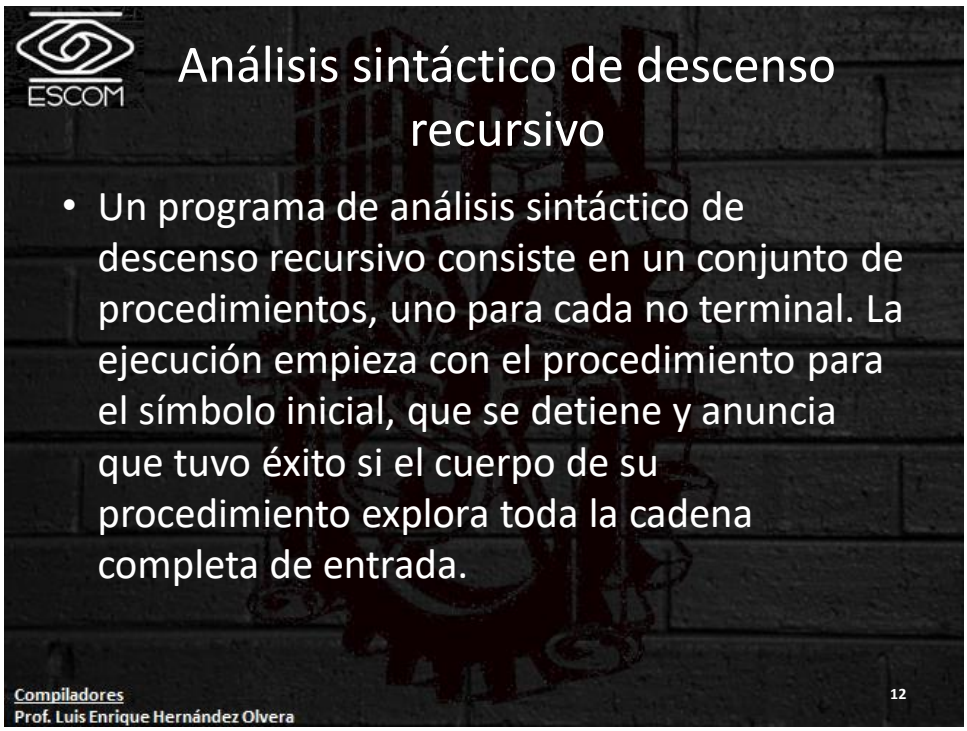
Comunicación del Analizador Sintáctico



Análisis Sintáctico

- Existen tres tipos generales de analizadores para las gramáticas: universales, descendentes y ascendentes.
- Los métodos universales de análisis sintáctico como el algoritmo de Cocke-Younger-Kasami y el algoritmo de Earley pueden analizar cualquier gramática, sin embargo, estos métodos generales son demasiado ineficientes como para usarse en la producción de compiladores.


ESCOM
Análisis Sintáctico
• Los métodos que se utilizan, por lo regular, en los compiladores pueden clasificarse como descendentes o ascendentes.
• Los métodos descendentes construyen árboles de análisis sintáctico de la parte superior (raíz) a la parte inferior (hojas), mientras que los métodos ascendentes empiezan de las hojas y avanzan hasta la raíz. En cualquier caso, la entrada al analizador se explora de izquierda a derecha, un símbolo a la vez.
Compiladores
Prof. Luis Enrique Hernández Olvera
11


ESCOM
Análisis sintáctico de descenso recursivo
• Un programa de análisis sintáctico de descenso recursivo consiste en un conjunto de procedimientos, uno para cada no terminal. La ejecución empieza con el procedimiento para el símbolo inicial, que se detiene y anuncia que tuvo éxito si el cuerpo de su procedimiento explora toda la cadena completa de entrada.
Compiladores
Prof. Luis Enrique Hernández Olvera
12

Análisis sintáctico de descenso recursivo

- Observe que este seudocódigo es no determinista, ya que empieza eligiendo la producción *A* que debe aplicar de una forma no especificada.

```
void A()
{
1) Elegir una producción A, A → X1X2 ... Xk ;
2)    for ( i = 1 a k ) {
3)       if ( Xi es un no terminal )
4)       llamar al procedimiento Xi();
5)       else if ( Xi es igual al símbolo de entrada actual a )
6)       avanzar la entrada hasta el siguiente símbolo;
7)       else /* ha ocurrido un error */;
      }
}
```


Análisis sintáctico de descenso recursivo

- El descenso recursivo general puede requerir de un rastreo hacia atrás; es decir, tal vez requiera exploraciones repetidas sobre la entrada. Sin embargo, raras veces se necesita el rastreo hacia atrás para analizar las construcciones de un lenguaje de programación, por lo que los analizadores sintácticos como éste no se ven con frecuencia. Incluso para situaciones como el análisis sintáctico de un lenguaje natural, el rastreo hacia atrás no es muy eficiente, por lo cual se prefieren métodos tabulares.