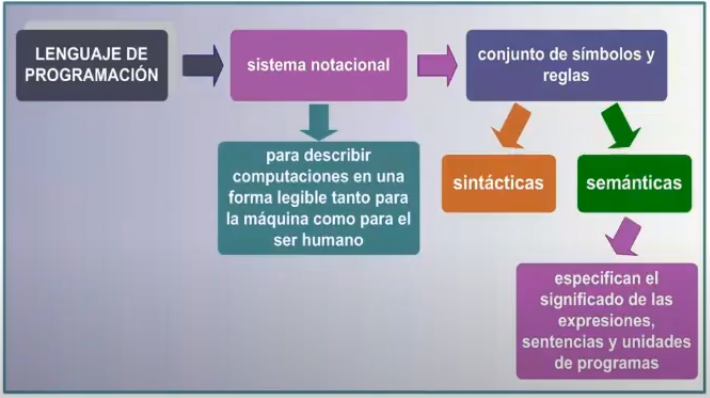
**Semántica**



**Concepto de semántica**

* Especifica el significado de las expresiones, sentencias y unidades de programa.
* Describir el significado de los programas sintácticamente validos escritos en un LP.
* El limite exacto entre los términos sintaxis y semántica es algo arbitrario y los autores lo ubican usando criterios diferentes. Un ejemplo puede ser considerar sintáctico todo aquellos aspectos que se resuelven en compilación y semánticos todos los demás.
* Todos los aspectos tenidos en cuenta durante la compilación con excepción de la generación de código y su optimización, puede ser considerado sintáctico, mientras que todas las cuestiones que se resuelven en ejecución son de naturaleza semántica.

**Elementos de la semántica de un LP (Desde el punto de vista del diseñador del lenguaje)**

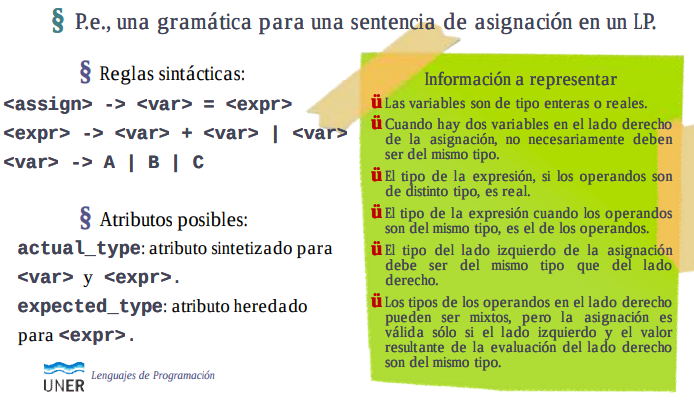
* Variables:
  + Corresponde a una dirección de memoria que se utiliza para almacenar valores que va a manipular el programa.
  + Se las va a referenciar por su nombre.
  + Por medio de las reglas sintácticas podemos saber como pueden nombrarse las variables.
  + Una declaración incluye una variable en un programa, le da un nombre y permite declarar algunas propiedades semánticas como por ejemplo el tipo de dato.
* Valores y referencias:
  + ¿Que valor se asocia con una variable?
    - L-Value:
      * Valor que denota una locación de memoria.
      * Puede usarse del lado izquierdo de una sentencia de asignación.
    - R-Value:
      * Valor que denota el contenido de una locación de memoria.
      * Puede usarse del lado derecho de una sentencia de asignación.
    - Por ejemplo en la sentencia **a = b** tenemos un R-Value para **b** (valor, el contenido de esa región de memoria) y un L-Value para **a** (referencia a la posición de memoria del valor **b**).
* Expresiones:
  + Las expresiones son estructuras sintácticas que permiten combinar valores y operaciones para calcular nuevos valores.
  + El LP especifica las reglas sintácticas y semánticas para construir las expresiones, dependiendo del LP puede obligarse a una expresión a producir un valor de un tipo o de tipos diferentes en momentos diferentes.
  + Ej:
    - request == 1 es una expresión del tipo booleano.
    - “invalid request .\ n” es una expresión del tipo array de caracteres.
  + El orden en el que se realizan las operaciones dentro de una expresión puede alterar su valor. Algunos lenguajes especifican estrictamente el orden en el que deben realizarse mediante las reglas de precedencia y asociatividad. Otros lenguajes dejan que el implementador (del traductor/compilador) decida en que orden se va a realizar la operación dentro de una expresión.
  + El problema semántico que existe alrededor de estas expresiones esta dado por el tipo de expresiones que permite el LP y sobre todo cuando este permite que una expresión produzca ciertos L-Valores que puedan ser usado como referencia.

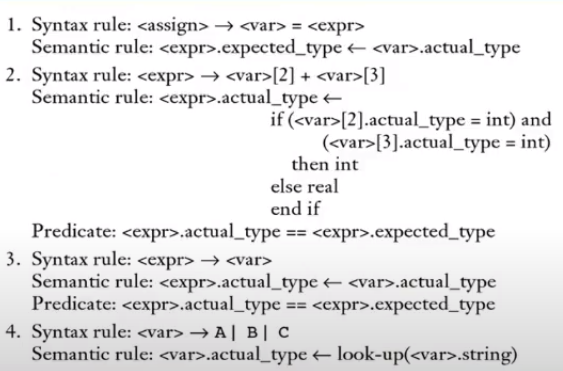
**Semántica formal**

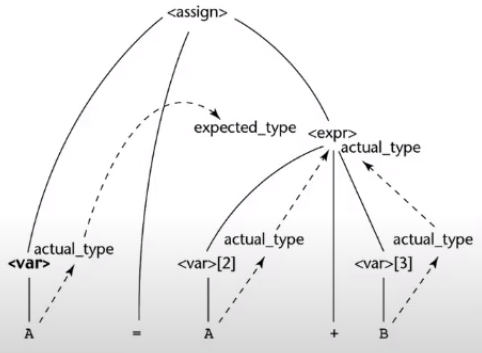
* Descripción rigurosa del significado o comportamiento de las construcciones de un LP.
* La especificación formal de la semántica de un LP deja en claro el comportamiento de ese LP.
* Una especificación completa tanto de la sintaxis como de la semántica puede ser usada como una herramienta que permita generar compiladores para un LP de manera automática, como también analizadores léxicos, sintácticos y semánticos.
* Importancia de una definición formal de la semántica de un LP radica en:
  + Facilitar a los diseñadores del lenguaje una forma no ambigua y consistente de definir un lenguaje, como así también les permite descubrir irregularidades, omisiones en el LP que están diseñando.
  + Estandarizar un LP por medio de una definición semántica no ambigua para ese lenguaje no va a permitir que existan variaciones entre implementación e implementación. Va a ser mas fácil estandarizar ese lenguaje.
  + Para los usuarios facilita la comprensión del lenguaje por parte del programador y como una forma de comunicación entre el diseñador, el implementador y el programador.
  + Permite/proporciona los fundamentos para pruebas de correctitud tanto de traductores como de programas e información relacionada con su ejecución.
* Para la especificación formal de la semántica de un LP, no hay un método estándar como si lo es BNF para la sintaxis.
* Una clasificación posible sería:
  + Semántica Estática:
    - Hay características de los LP que no pueden describirse con gramáticas libre de contexto, pero pueden resolverse en tiempo de compilación. La Semántica Estática tiene que ver con los aspectos que no pueden ser expresados por la BNF, pero que pueden evaluarse en tiempo de compilación. Están relacionados con las formas o estructuras validas y el análisis para el chequeo de tipos por ejemplo pueden hacerse en tiempo de compilación.
    - Para describir esta semántica formalmente se utiliza la gramática de atributos, inventada por Knuth en los años 60
  + Semántica Dinámica:
    - Describe el significado de las expresiones, de las sentencias, de las unidades de programa en ejecución.
    - Describe el efecto de ejecutar las diferentes construcciones en el LP y los programas solo se pueden ejecutar si son correctos para la semántica estática y para la sintaxis.
    - Para describirla se puede utilizar:
      * Semántica Operacional: de utilidad para el implementador del LP, dado que necesita conocer que significan exactamente las estructuras para diseñar una implementación correcta y mostrar que esas estructuras producen programas que muestran el comportamiento que se ha especificado en la definición.
      * Semántica Axiomática: de utilidad para el programador, necesita conocer el significado de las sentencias de un LP antes de poder usarlas.
      * Semántica Denotacional: es utilizada por el diseñador para descubrir ambigüedades o inconsistencias que puedan presentarse en el LP que esta diseñando.

**Semántica Estática**

* Gramática de atributos:
  + Asignan propiedades o atributos a distintas estructuras sintácticas del LP. Estos atributos pueden describir información semántica que permite implementar un analizador semántico, Ej. el tipo de una expresión, o para representar cualquier otra propiedad dado que no están relacionados con el comportamiento de los programas en ejecución.
  + Es una gramática libre de contexto (la que nos permitió definir la sintaxis del LP) a la que se le agrega los siguientes elementos que van a llevar información semántica por los nodos del árbol de análisis sintáctico:
    - Atributos:
      * Propiedad de una estructura sintáctica del lenguaje.
      * Son valores asociados a cada símbolo de la gramática, ya sean terminales o no terminales.
      * Varían según la información que contienen y de la etapa del procesamiento. Si están calculados en la fase de traducción por un compilador o en ejecución por un interprete.
      * Ej. El tipo de una variable, valor de una expresión, dirección de memoria, etc.
      * El conjunto de atributos asociados a cada símbolo de la gramática esta formado por dos conjuntos disjuntos:
        + Conjunto de atributos sintetizados S(x) : La información semántica que describen es pasada de abajo hacia arriba entre los nodos del árbol sintáctico.
        + Conjunto de atributos heredados I(x): la información semántica fluye de arriba hacia abajo por los nodos de ese árbol.
      * Atributos intrínsecos: conjunto particular de atributos sintetizados asociados a las hojas del árbol, no a los nodos y cuyo valor se calcula fuera del árbol sintáctico.
      * Reglas semánticas:
        + Conjunto de funciones para definir y calcular los valores de los atributos definidos que están asociados a las reglas de la gramática.
        + Establecen relaciones entre los valores de los atributos de una gramática de atributos expresadas mediante una función matemática.
      * Condiciones:
        + Conjunto de predicados para evaluar la consistencia de los atributos asociados a las reglas de la gramática.
        + Cada símbolo no terminal de una gramática libre de contexto puede ser asociada a un conjunto finito de atributos. Cada atributo puede ser referenciado a través de un nombre y tiene asociado un dominio de valores.
        + Un atributo puede estar vinculado a varios símbolos no terminales y cada aparición de este puede estar ligada a una condición lógica que exprese una restricción que debe ser satisfecha por el valor de ese atributo.
        + El valor de un atributo se va a calcular a través de las reglas de evaluación asociadas con la regla de producción.

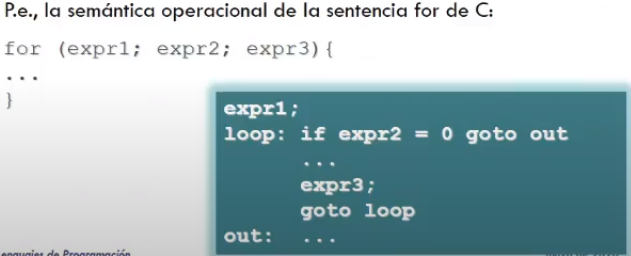




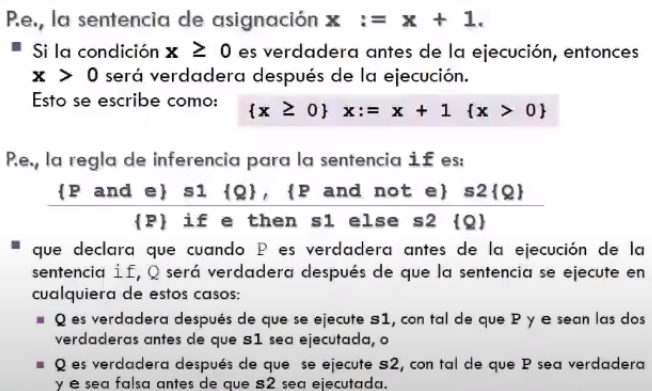


**Semántica Dinámica**

* Semántica Operacional:
  + Describe el significado de un programa ejecutando sus sentencias en una maquina, a través de un lenguaje de bajo nivel. Los cambios de estado que se producen en esa maquina cuando se ejecuta una sentencia del LP definen el significado de esa sentencia.
  + Frecuentemente usada en textos de programación y en los manuales de referencia del lenguaje.
  + Muy cercana a la implementación, se puede usar para construir prototipos de procesadores de LP.



* + Cada una de las sentencias del lenguaje de alto nivel se describe por medio de un lenguaje de bajo nivel en términos de saltos condicionales e incondicionales.
  + La desventaja que tiene la semántica operacional al utilizar otro lenguaje de bajo nivel para describir la semántica de un LP, algunos conceptos pueden definirse en términos de ellos mismos.
* Semántica Axiomática:
  + Describe las acciones de un programa determinando predicados lógicos que especifican restricciones sobre las variables del programa antes y después de su ejecución. Existe un estado anterior y posterior a la ejecución del constructor.
  + Es utilizada para la verificación formal de los programas.
  + Basada en la lógica formal (calculo de predicados) y los axiomas son definidos para cada tipo de sentencia y permiten transformación de expresiones lógicas en otras expresiones lógicas.
  + Aserciones:
    - Son los predicados.
    - Expresiones lógicas usadas en la semántica axiomática.
    - A cada sentencia del programa le precede y le sigue una expresión lógica cuyo objetivo es especificar cual es el significado de esa sentencia.
    - Una aserción antes de una sentencia es una precondición (afirman las relaciones entre las variables y las restricciones que se cumplen en ese momento en la ejecución).
    - Una aserción después de una sentencia es una postcondición.
    - Esto normalmente se escribe {P} S {Q}, indica que si la sentencia es ejecutada en un estado en el cual la aserción P o precondición se satisfacen entonces resultara en un estado en el cual se va a satisfacer la aserción Q.
    - La condición previa mas débil es la menos restrictiva que garantizara la validez de la condición posterior o postcondición a asociada.
    - Regla de inferencia:
      * Es un método para inferir la verdad de una afirmación sobre la base de los valores de otras afirmaciones. En general se escribe de la forma antecedente sobre consecuente.
    - Axioma:
      * Declaración lógica que se supone que es verdadera.
      * Regla de inferencia sin antecedente.



* Semántica Denotacional:
  + Define el comportamiento de un LP aplicando funciones matemáticas a programas o a construcciones del lenguaje, para representar su significado.
  + Es el método mas riguroso y ampliamente conocido para describir el significado de los programas o construcciones en un LP donde se define una correspondencia entre los constructores sintácticos y el significado.
  + Se puede predecir el comportamiento de un programa sin necesidad de ejecutarlo en una computadora. Brindar información sobre el efecto de la ejecución del programa y no como se va a llevar a cabo.
  + Este tipo de semántica se utilizo para especificar la semántica de lenguajes como ALGOL, ADA y Pascal. El proceso de construcción de una especificación semántica denotacional requiere que se defina para cada entidad del LP un objeto matemático y una función que va a mapear instancias de esas entidades/construcciones del LP en instancias del objeto matemático.
  + Basada en la teoría de funciones recursivas, donde las funciones tienen un dominio y un rango.
    - Dominio sintáctico: esta dado por las estructuras sintácticas del lenguaje que se van a mapear.
    - Dominio semántico: es la colección de objetos con los que se van a mapear los parámetros. Rango de la función.

