



PROCESADORES DE LENGUAJE

Tema 3: Traducción dirigida por sintaxis

Objetivos del tema

- Objetivos
 - Entender el concepto “traducción dirigida por la sintaxis”
 - Aprender a especificar gramáticas con atributos
 - Entender el concepto de reglas semánticas
 - Entender los problemas de dependencias
 - Aprender a distinguir y reconocer
 - Compiladores de una sola pasada
 - Compiladores de doble pasada

Contenido

- Definiciones dirigidas por la sintaxis.
- Gramáticas de atributos.
- Traducción dirigida por sintaxis.
- Atributos sintetizados y heredados. Grafos de atributos.
- Evaluación de atributos durante el análisis descendente.

Bibliografía recomendada

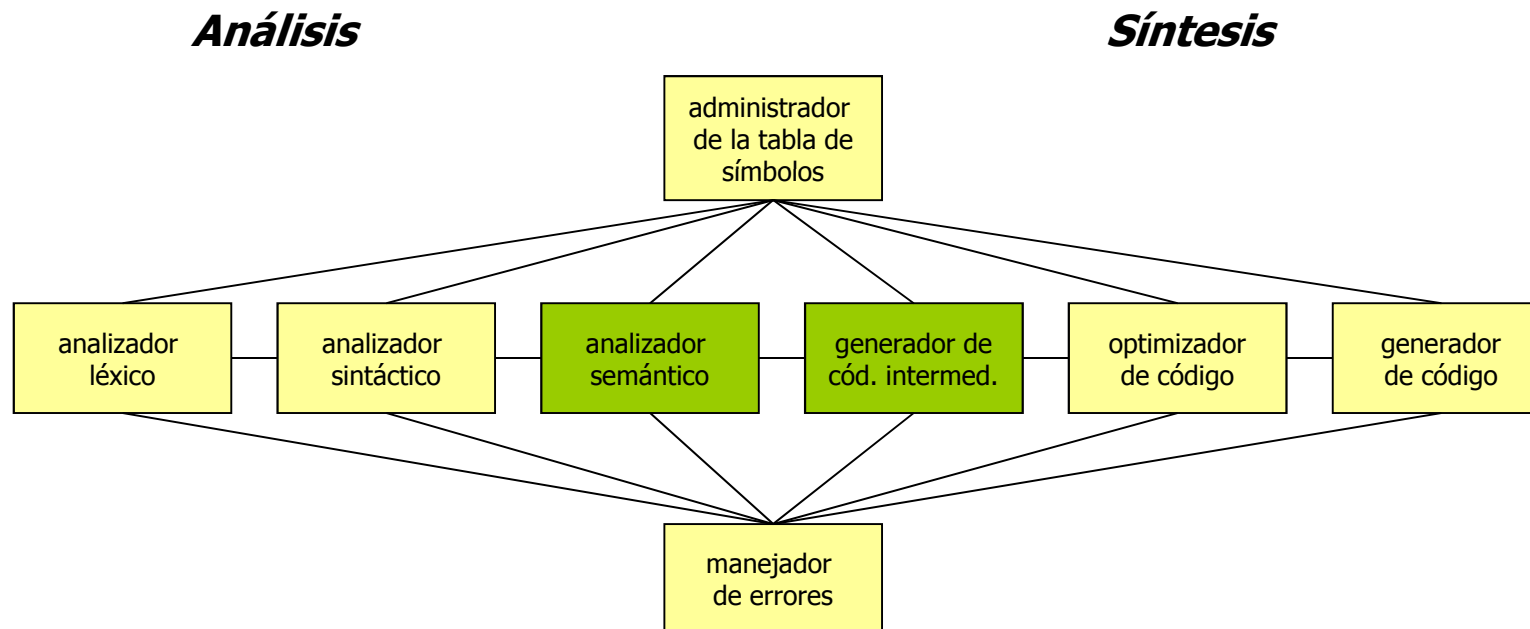
- **Compiladores** (Principios, técnicas y herramientas)
PEARSON, Addison Wesley Longman
 - Capítulo 5: Traducción dirigida por la sintaxis

- **Diseño de compiladores**
A. Garrido, J. Iñesta, F. Moreno y J. Pérez. 2002.
Universidad de Alicante.

1. Definiciones dirigidas por la sintaxis

Introducción

- Estructura de un compilador: análisis semántico y generación de código intermedio.



1. Definiciones dirigidas por la sintaxis

¿Qué es la semántica de un programa?

- Sintaxis
 - Cómo se ve un programa
 - Representación textual o estructura
 - Es posible dar una definición precisa

- Semántica
 - Cuál es el significado del programa
 - Es más difícil dar una definición precisa

1. Definiciones dirigidas por la sintaxis

Análisis Semántico

- **Semántica:** conjunto de reglas que especifican el significado de cualquier sentencia sintácticamente correcta y escrita en un determinado lenguaje.
- El **análisis semántico**, a diferencia de otras fases, no se realiza claramente diferenciado del resto de las tareas del compilador.
 - Fase en la que se obtiene información necesaria para la compilación tras conocer la estructura sintáctica del programa.
 - Completa las fases de análisis léxico y sintáctico incorporando comprobaciones que no pueden asimilarse al mero reconocimiento de una cadena dentro de un lenguaje.
 - La información que usa para decidir si una sentencia es correcta es útil también para generar código: por eso muchas veces las 2 tareas se hacen 'juntas'.

1. Definiciones dirigidas por la sintaxis

Análisis Semántico

- Además de comprobar que un programa cumple con las reglas de la gramática, hay que comprobar que lo que se quiere hacer tiene sentido.
- Esta fase también modifica la tabla de símbolos y suele estar mezclada con la generación de código intermedio.
- Las gramáticas independientes del contexto no son suficientes para realizar el análisis semántico: Por ejemplo, no hay forma de comprobar si una variable ha sido definida ya, o si existe una determinada etiqueta.
- Es necesario definir un tipo de **gramática más rica como las gramáticas de atributos.**

1. Definiciones dirigidas por la sintaxis

Análisis Semántico

- Errores semánticos de un programa:
 - Conversiones de tipos no permitidas

```
int x;
```

```
x = 4.32;
```

```
Error: Ej1.java [6:1] possible loss of precision
```

- Variables usadas y no definidas
 - Operandos de tipos no compatibles

```
if (x || 5) x = 0;
```

```
Error: Ej2.java [7:1] operator || cannot be applied to int,int
```

1. Definiciones dirigidas por la sintaxis

Funciones del análisis semántico

- Identificar cada tipo de instrucción y sus componentes.
- Completar la **Tabla de Símbolos**.
- Realizar comprobaciones *estáticas*:
 - Se realizan durante la compilación del programa.
 - Ejemplos: comp. de tipos, unicidad de etiquetas e identificadores, declaraciones de las variables, etc.
- Realizar comprobaciones *dinámicas*:
 - Aquellas que el compilador incorpora al programa traducido.
 - Hacen referencia a aspectos que sólo pueden ser conocidos en tiempo de ejecución
 - Dependientes del estado de la máquina en la ejecución o del propio programa.
 - Ejemplos: división por cero, desbordamiento, fallo en apertura ficheros, error en dispositivo, índices en arrays...

1. Definiciones dirigidas por la sintaxis

Funcionamiento del análisis semántico

- Recibe como entrada el árbol de derivación del programa fuente (normalmente en su versión 'abstracta': AST o abstract syntax tree)
- Añade al árbol una serie de anotaciones, que permiten determinar la corrección semántica del programa y preparar la generación de código
- La salida que genera es o un árbol con anotaciones semánticas o directamente código.

1. Definiciones dirigidas por la sintaxis

Especificación de la semántica

- No hay una notación estándar para especificar la semántica estática de un lenguaje
 - El análisis semántico varía mucho de unos lenguajes a otros
- Las especificaciones semánticas de un lenguaje pueden hacerse de manera informal o formal:
 - Especificación natural: basada en el lenguaje natural.
 - Por ejemplo:
 - “Los identificadores deben definirse antes de utilizarse”
 - “Los operandos deben ser compatibles entre sí”
 - Especificación formal: definición más precisa.
 - Gramáticas de atributos (Knuth, 1968)

1. Definiciones dirigidas por la sintaxis

Tipos de análisis semánticos

- En **compiladores de un solo paso**, las comprobaciones semánticas se realizan directamente desde el analizador sintáctico y son dichas "rutinas" las que llaman al **generador de código**. El instrumento más utilizado para conseguirlo es la gramática de atributos.
- En **compiladores de dos o más pasos**, el análisis semántico se realiza independientemente de la generación de código, pasándose información a través de un archivo intermedio, que normalmente contiene información sobre el árbol sintáctico.
- En cualquier caso, las rutinas semánticas suelen hacer uso de una pila (la pila semántica) que contiene la información semántica asociada a los operandos (y a veces a los operadores) en forma de *registros semánticos*.

1. Definiciones dirigidas por la sintaxis

Tipos de análisis semánticos

- Compiladores de 1 pasada :
 - Toda la fase de síntesis del compilador se genera en las acciones semánticas.
 - más rápido, menos memoria en caso computador lento.
 - pueden condicionar definición de lenguajes,
 - por ej: procedimientos forward de Pascal.
- Compiladores más modernos, muchas pasadas (5-15) más modular, menos limitaciones
 - Entre las pasadas, el compilador genera representaciones intermedias, si usa k pasadas, entonces genera k-1 representaciones intermedias.

1. Definiciones dirigidas por la sintaxis

¿Cómo hace el analizador semántico su trabajo?

- Usa como herramienta las **gramáticas atribuidas**: se basa en el análisis sintáctico
- Gramáticas con atributos: reglas + símbolos de acción semántica, que son rutinas semánticas: se activan para realizar un análisis semántico y/o generar código.
- **Traducción dirigida por la sintaxis**: la semántica está ligada a la sintaxis. El significado de una frase está directamente relacionado con su estructura sintáctica según se representa en su árbol de análisis sintáctico
- Se realizan en paralelo:
 - Análisis semántico
 - Traducción

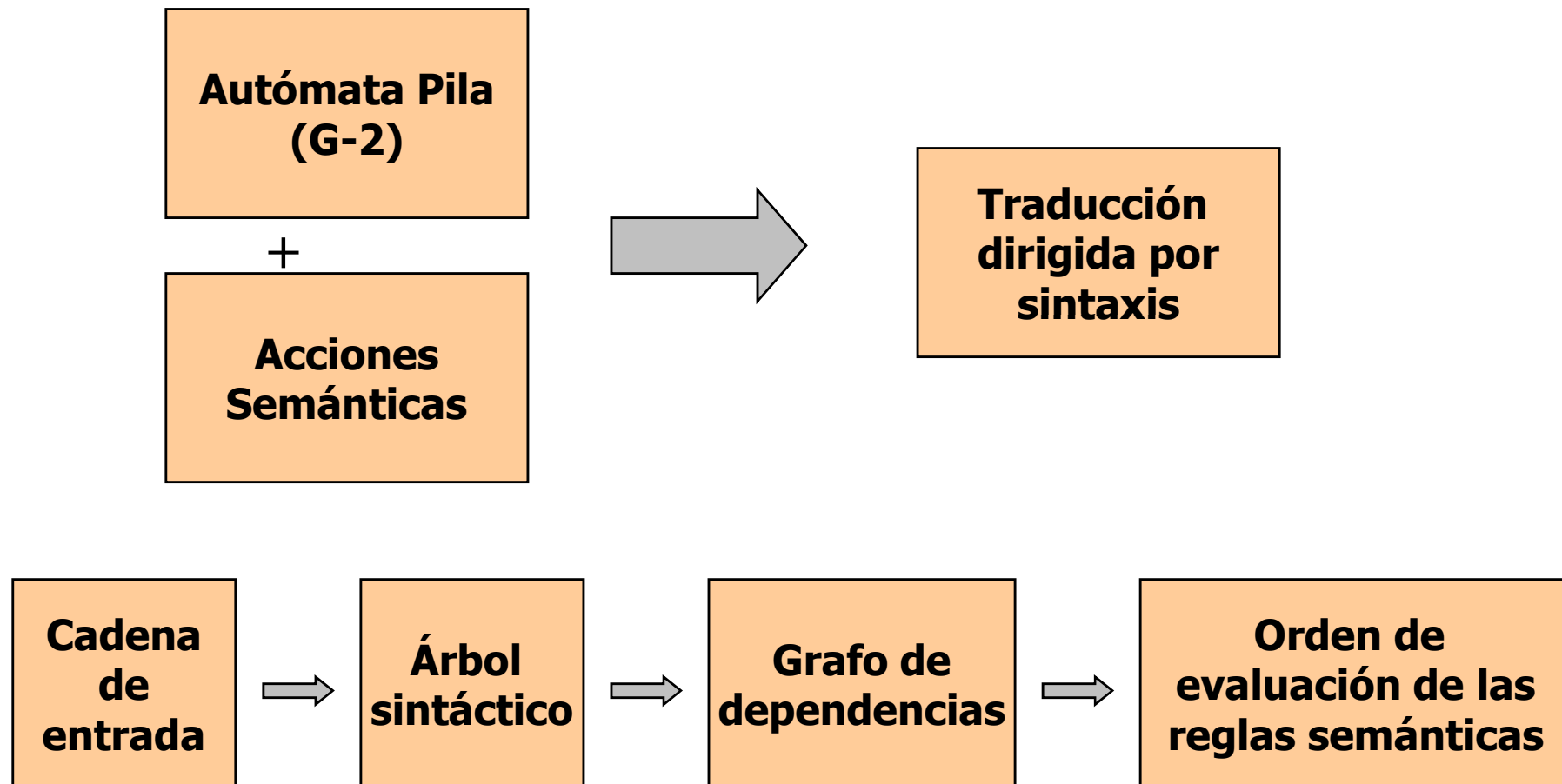
2. Gramáticas de atributos

¿Cómo hace el analizador semántico su trabajo?

- Una **gramática de atributos** es una **gramática libre de contexto** cuyos **símbolos** pueden tener **asociados atributos** y las **producciones** pueden tener **asociadas reglas de evaluación** de los atributos. Es una herramienta para el análisis semántico.
- Las reglas semánticas son un método para expresar la relación entre el cálculo de los atributos y las reglas del lenguaje.
- Cada producción (regla sintáctica) tiene asociada una o varias acciones semánticas.

2. Gramáticas de atributos

Traducción dirigida por sintaxis



2. Gramáticas de atributos

Traducción dirigida por sintaxis

- Hay 2 notaciones para asociar reglas semánticas a producciones: (también se usarán para la generación de código intermedio)
 - Definición dirigida por la sintaxis (DDS)
 - Esquema de Traducción (EDT)
- Con ambas notaciones, se analiza sintácticamente la cadena de entrada, se construye el árbol sintáctico y luego se recorre ese árbol para evaluar las reglas semánticas en sus nodos.
- En la evaluación de cada acción (para realizar alguna comprobación semántica),
 - Se generará código.
 - Se accederá a la tabla de símbolos para guardar/consultar información.
 - Se pueden notificar mensajes de error

2. Gramáticas de atributos

Para asociar reglas semánticas a producciones

- **Definiciones dirigidas por la sintaxis (DDS)**
 - Formalismo de alto nivel para describir traducciones
 - Se ocultan los detalles de implementación
 - No se impone orden de ejecución de las reglas

- **Esquemas de traducción (ETDS)**
 - Notación de bajo nivel para especificar un traductor
 - El traductor es de una sola pasada
 - Explícita el orden de la ejecución de las acciones

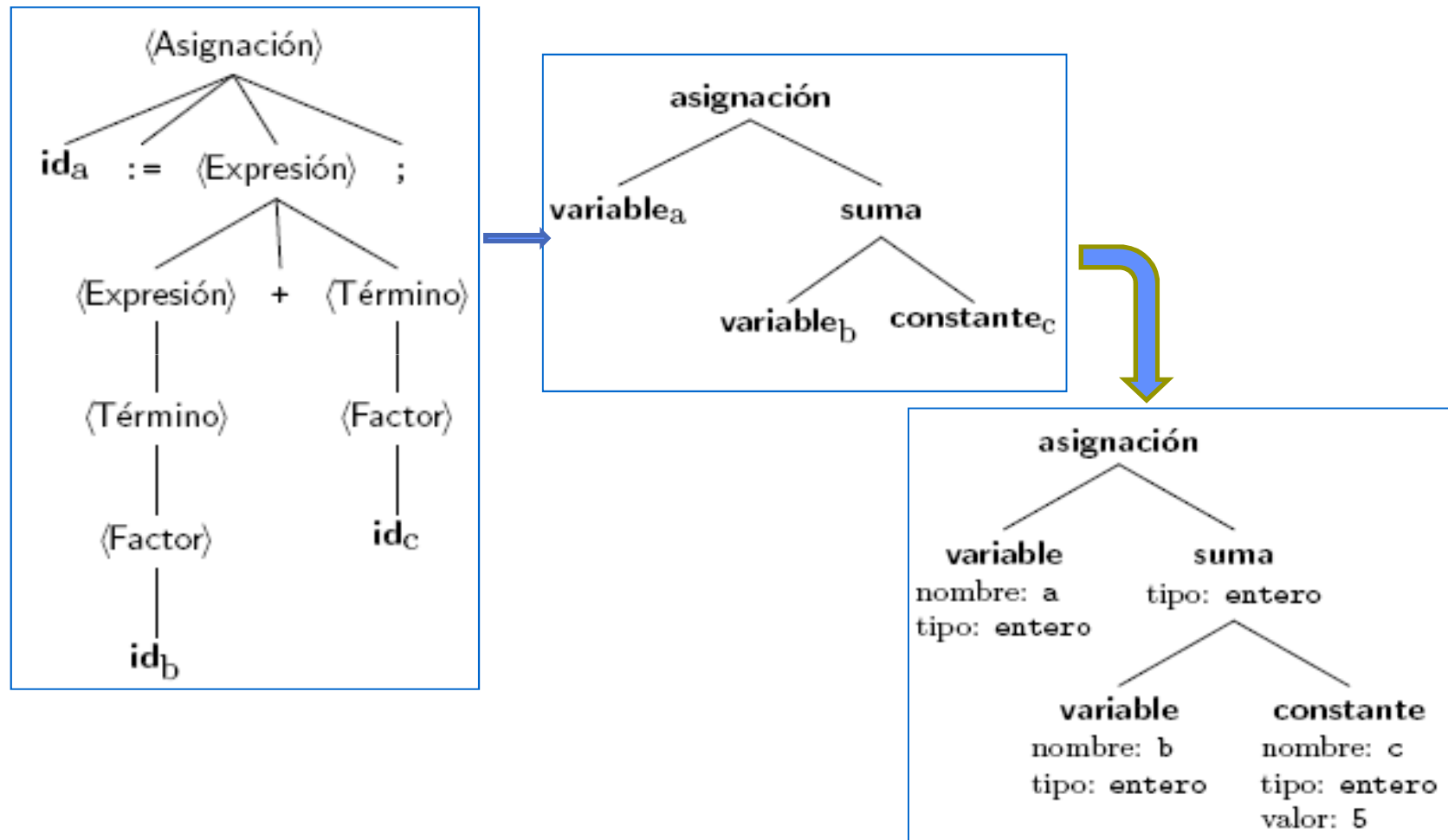
2. Gramáticas de atributos

El árbol sintáctico abstracto

- **Representación compacta** (sin redundancia) correspondiente a una gramática, caracterizada por que los nodos interiores son operadores, en sentido amplio.
- Útil para ser “anotado” o “decorado” con atributos en recorridos interiores, y a partir de ahí:
 - Realizar chequeos
 - Generar código (intermedio)
- Durante el análisis semántico se recoge una serie de informaciones que resultan de utilidad para fases posteriores. Estas informaciones se pueden almacenar en el árbol, “decorándolo”:

2. Gramáticas de atributos

El árbol sintáctico abstracto



2. Gramáticas de atributos


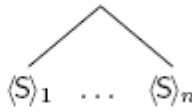
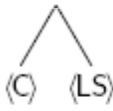

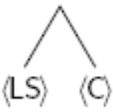
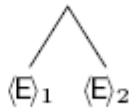
El árbol sintáctico abstracto

- Es similar a los árboles de análisis sintáctico, pero tiene algunas diferencias importantes:
 - No aparecen todos los componentes léxicos del programa. Por ejemplo:
 - No es necesario incluir los paréntesis de las expresiones.
 - No se necesitan los separadores o terminadores de las sentencias.
- Pueden aparecer otros componentes no estrictamente sintácticos, como acciones de coerción de tipos.

2. Gramáticas de atributos

Construcción de árbol sintáctico abstracto (AST)

- Definiremos qué elementos emplear para cada estructura:

Estructura	Representación	Estructura	Representación
if $\langle E \rangle$ then $\langle LS \rangle$ end	<p>si</p> 	begin $\langle S \rangle_1 \dots \langle S \rangle_n$ end	<p>sentencias</p> 
while $\langle C \rangle$ do $\langle LS \rangle$ end	<p>mientras</p> 	id := $\langle E \rangle$;	<p>asignación</p> 
repeat $\langle LS \rangle$ until $\langle C \rangle$;	<p>repetir</p> 	$\langle E \rangle_1 + \langle E \rangle_2$	<p>suma</p> 
...

- Luego usaremos los atributos para construir el árbol, usando el atributo arb para devolver el árbol que construye cada no terminal, podemos hacer algo parecido a:

2. Gramáticas de atributos

Construcción de árbol sintáctico abstracto (AST)

$\langle \text{Sentencia} \rangle \rightarrow \text{if } \langle \text{Expresión} \rangle \text{ then } \langle \text{Sentencias} \rangle \text{ end}$
 $\{\langle \text{Sentencia} \rangle.\text{arb} := \text{NodoSi}(\langle \text{Expresión} \rangle.\text{arb}, \langle \text{Sentencias} \rangle.\text{arb})\}$

$\langle \text{Sentencia} \rangle \rightarrow \text{while } \langle \text{Expresión} \rangle \text{ do } \langle \text{Sentencias} \rangle \text{ end}$
 $\{\langle \text{Sentencia} \rangle.\text{arb} := \text{NodoMientras}(\langle \text{Expresión} \rangle.\text{arb}, \langle \text{Sentencias} \rangle.\text{arb})\}$

$\langle \text{Sentencia} \rangle \rightarrow \text{repeat } \langle \text{Sentencias} \rangle \text{ until } \langle \text{Expresión} \rangle ;$
 $\{\langle \text{Sentencia} \rangle.\text{arb} := \text{NodoRepetir}(\langle \text{Sentencias} \rangle.\text{arb}, \langle \text{Expresión} \rangle.\text{arb})\}$

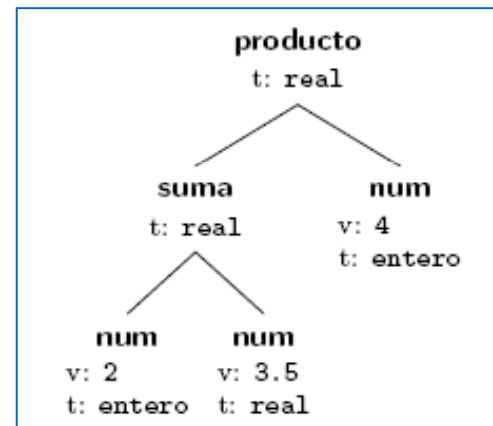
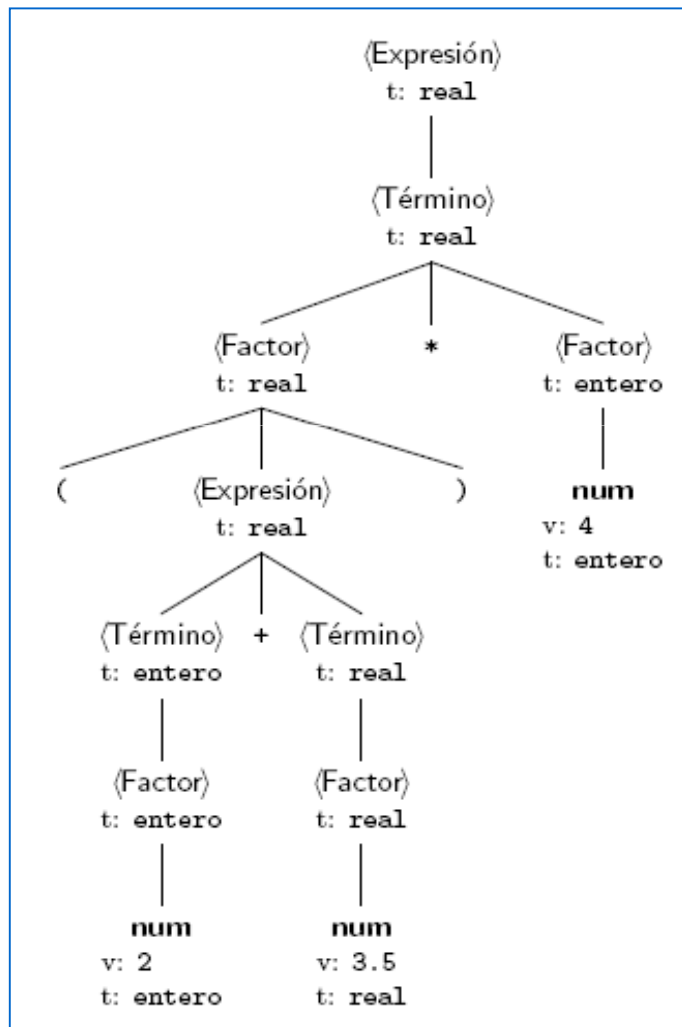
$\langle \text{Sentencia} \rangle \rightarrow \text{id} := \langle \text{Expresión} \rangle ;$
 $\{\langle \text{Sentencia} \rangle.\text{arb} := \text{NodoAsignación}(\text{id.lexema}, \langle \text{Expresión} \rangle.\text{arb})\}$

$\langle \text{Sentencias} \rangle \rightarrow \{l := \lambda\} (\langle \text{Sentencia} \rangle \{l := l + \langle \text{Sentencia} \rangle.\text{arb}\})^*$
 $\{\langle \text{Sentencias} \rangle.\text{arb} := \text{NodoSentencias}(l)\}$

$\langle \text{Expresión} \rangle \rightarrow \langle \text{Término} \rangle_1 \{\text{arb} := \langle \text{Término} \rangle_1.\text{arb}\}$
 $(+ \langle \text{Término} \rangle_2 \{\text{arb} := \text{NodoSuma}(\text{arb}, \langle \text{Término} \rangle_2.\text{arb})\})^*$
 $\{\langle \text{Expresión} \rangle.\text{arb} := \text{arb}\}$

2. Gramáticas de atributos

Evaluación de atributos sobre el AST



La elección a cerca de si evaluar atributos en el AST o durante el A.Sint. es básicamente una cuestión de simplicidad. Dependerá del tipo de atributos y del tipo de análisis sintáctico realizado.

2. Gramáticas de atributos

Conclusión

- Así, el **objetivo** de la fase de análisis semántico será **doble**: por un lado **detectaremos errores** que no se han detectado en fases previas y por otro lado **obtendremos el AST** decorado de la entrada.
- Para ello **utilizaremos Definiciones dirigidas por la sintaxis** o esquemas de traducción, que permitirán asociar acciones a las reglas de la gramática.
- Estas acciones realizarán comprobaciones y construirán el AST que después se recorrerá para terminar las comprobaciones y será la base para la interpretación o la generación de código.

3. Traducción dirigida por la Sintaxis

Gramáticas de atributos

- **Atributo:** propiedad de una construcción de un lenguaje.
 - Pueden variar mucho en cuanto a información que contienen o tiempo que tardan en determinarse durante la traducción/ejecución.
 - Cada símbolo (terminal o no terminal) puede tener asociado un número finito de atributos.
- Ejemplos de atributos:
 - Tipo de una variable
 - Valor de una expresión
 - Ubicación en memoria de una variable
 - Código objeto de un procedimiento
 - Número de dígitos significativos en un número

3. Traducción dirigida por la Sintaxis

Gramáticas de atributos

- Generalmente se denotan mediante un nombre precedido por un punto y el nombre del símbolo al que están asociados.

`NombreSímbolo.NombreAtributo`

- Ejemplo de uso (evaluación de atributos):

`numero → numero digito | digito`

a) `numero → digito`

`numero.valor = digito.valor`

b) `numero → numero digito`

`numero1.valor = numero2.valor * 10 +
digito.valor`

3. Traducción dirigida por la Sintaxis

Gramáticas de atributos

- Ejemplo de evaluación de atributos:

```
Exp → Exp op_aritmetica Exp  {  
    si ($1.tipo == $3.tipo) entonces  
        $$ .tipo = $1.tipo  
    si no  
        $$ .tipo = ERROR  
        Escribir("error tipos incompatibles")  
    fin_si  
}
```

3. Traducción dirigida por la Sintaxis

Gramáticas de atributos

```
<Expresion> ::= <Expresion> <Operador> <Expresion> {  
    <Operador>.Tipo = Mayor_tipo(<expresion2>.Tipo,<expresion3>.Tipo)  
    <Expresion1>.Tipo = <Operador>.Tipo  
    if    (<Operador>.Tipo == 'F' && <Expresion2>.Tipo == 'I'){  
        <Expresion2>.Tipo = 'F';  
        <Expresion2>.Valor = Float(<Expresion2>.Valor);  
    }  
    if    (<Operador>.Tipo == 'F' && <Expresion3>.Tipo == 'I'){  
        <Expresion3>.Tipo = 'F';  
        <Expresion3>.Valor = Float(<Expresion3>.Valor);  
    }  
    switch (<Operador>.Tipo){  
        'I':    <Expresion1>.Valor = Op_entera(<operador>.Clase,  
            <Expresion2>.Valor, <Expresion3>.Valor); break;  
        'F':    <Expresion1>.Valor = Op_real(<operador>.Clase,  
            <Expresion2>.Valor, <Expresion3>.Valor); break;  
    }  
}
```

3. Traducción dirigida por la Sintaxis

Gramáticas de atributos

- Las gramáticas de atributos se escriben en forma de tabla:
 - Las reglas gramaticales, a la izquierda
 - Las reglas semánticas asociadas, a la derecha

Regla gramatical	Regla semántica
Regla 1	Ecuaciones de atributo asociadas
...	...
Regla n	Ecuaciones de atributo asociadas

3. Traducción dirigida por la Sintaxis

Gramáticas de atributos

- Ejemplo:

Regla gramatical	Regla semántica
$L \rightarrow E$	<code>print(E.val)</code>
$E \rightarrow E + T$	<code>E₀.val = E₁.val + T.val</code>
$E \rightarrow T$	<code>E.val = T.val</code>
$T \rightarrow T * F$	<code>T₀.val = T₁.val * F.val</code>
$T \rightarrow F$	<code>T.val = F.val</code>
$F \rightarrow (E)$	<code>F.val = E.val</code>
$F \rightarrow \text{digito}$	<code>F.val = digito.valor_lexico</code>

3. Traducción dirigida por la Sintaxis

Evaluación de atributos

- Cada símbolo gramatical tiene asociado un conjunto de atributos.
- El valor de un atributo en un árbol sintáctico se calcula mediante una regla semántica asociada a la producción utilizada en el nodo.
- Tipos de atributos:
 - **Sintetizados:** Su valor se calcula en función de atributos de nodos hijos en el árbol de análisis sintáctico.

$$A \rightarrow aB \quad \{ A.\text{atributo} = a.\text{atributo} + B.\text{atributo} \}$$

- **Heredados:** Para un hijo se calculan a través de los atributos del padre y hermanos en el árbol de análisis sintáctico.

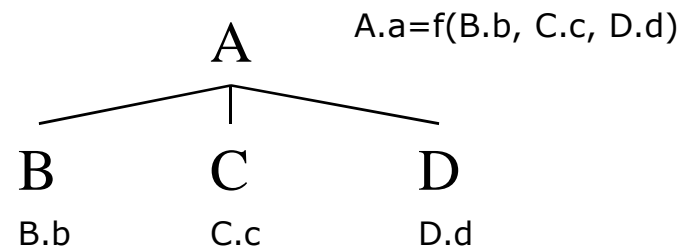
$$A \rightarrow aB \quad \{ B.\text{atributo} = a.\text{atributo} - A.\text{atributo} \}$$

3. Traducción dirigida por la Sintaxis

Tipos de atributos

■ Atributos sintetizados

- Se calculan a partir de atributos de nodos hijos
- Se refieren a atributos de la parte izda de la regla
- La información asciende por el árbol (AS Ascendente).
- Los atributos sintetizados lo son durante toda la gramática



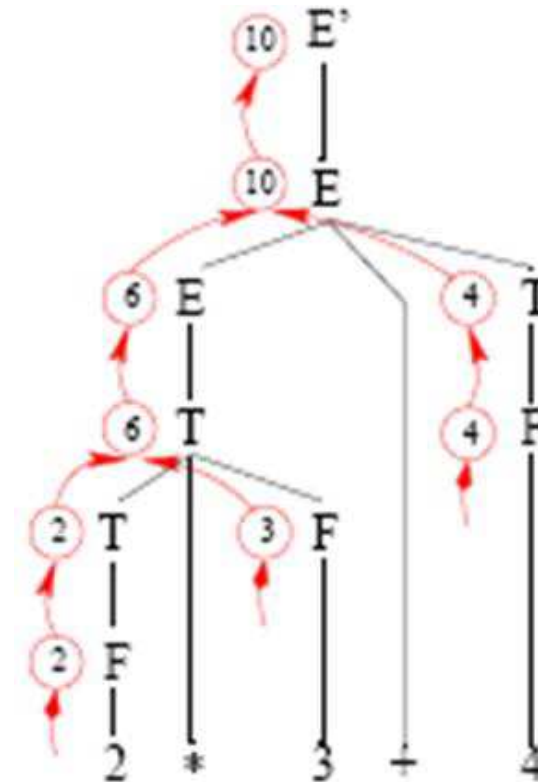
- **Gramáticas S-Atribuidas:** Cuando todos los atributos asociados con los símbolos gramaticales son sintetizados

3. Traducción dirigida por la Sintaxis

Ejemplo de Gr. Atrib. con atributos sintetizados

Todas las flechas que aparecen en la figura, que indican el orden de realización de las operaciones. En este caso son ascendentes, es decir: la información que se asigna a la parte izquierda de la regla se ha calculado utilizando únicamente información procedente de la parte derecha de la misma regla.

$E' \Rightarrow E$	$E'.val := E.val$
$E \Rightarrow E + T$	$E.val := E.val + T.val$
$E \Rightarrow T$	$E.val := T.val$
$T \Rightarrow T * F$	$T.val := T.val * F.val$
$T \Rightarrow F$	$T.val := F.val$
$F \Rightarrow (E)$	$F.val := E.val$
$F \Rightarrow a$	$F.val := LEXVAL(a)$

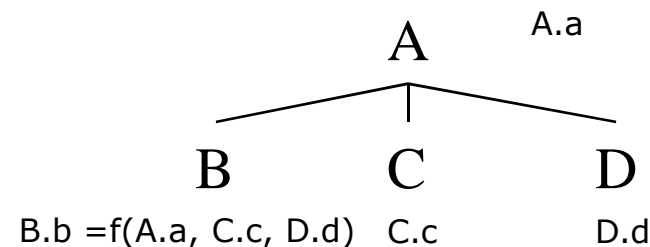


3. Traducción dirigida por la Sintaxis

Tipos de atributos

■ Atributos heredados

- Calculados con atributos en nodos padre y hermanos
- Expresan la dependencia de una construcción del leng. con el contexto en que aparece
- Se refieren a atributos del consecuente de la regla
- Información descendente o de tránsito horizontal
- Los atributos heredados lo son durante toda la gramática
- Se utilizan para transferir información entre reglas



3. Traducción dirigida por la Sintaxis

Tipos de atributos

- Ejemplo: atributo 'tipo' en esta gramática:

$Decl \rightarrow Type\ Var_List$

$Type \rightarrow int \mid float$

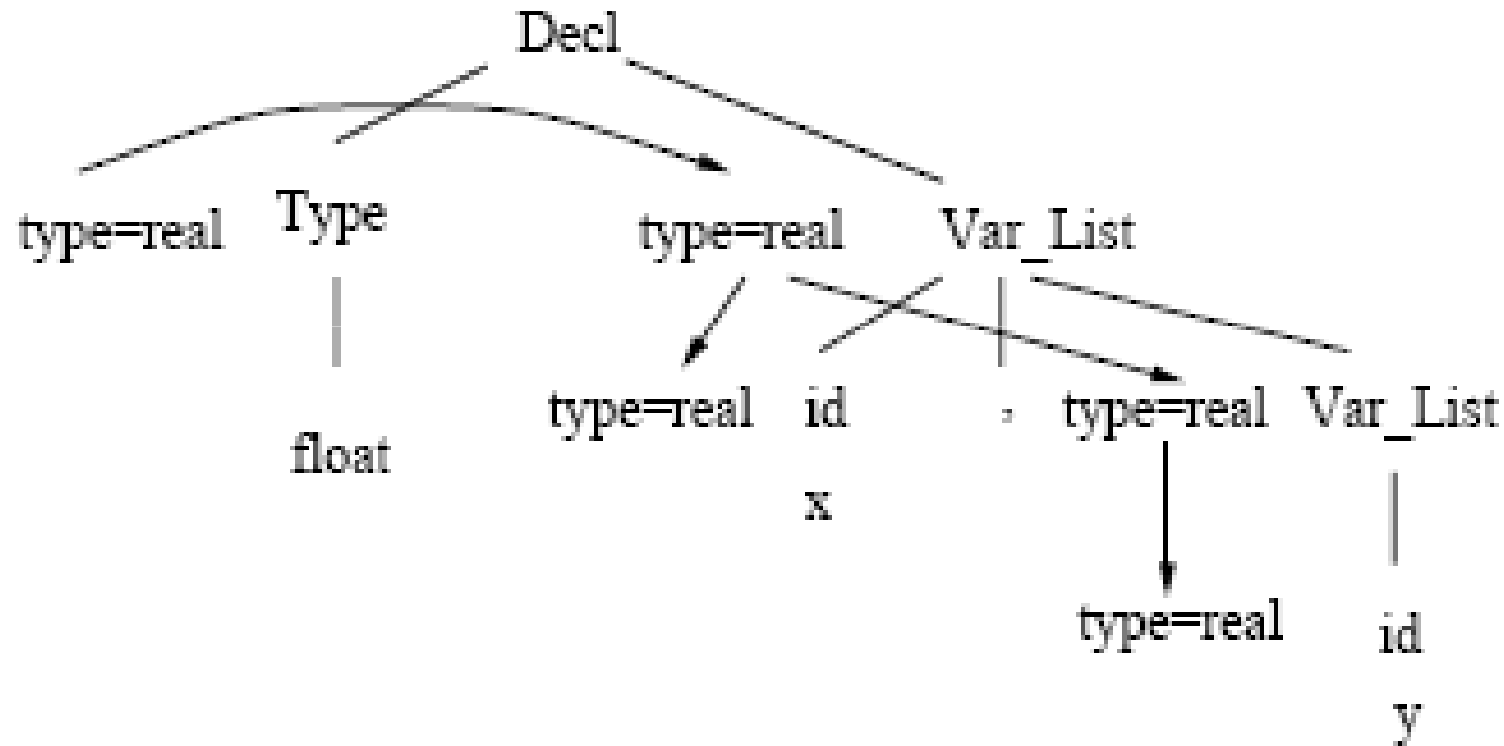
$Var_List \rightarrow id, Var_List \mid id$

Producción	Reglas semánticas
$Decl \rightarrow Type\ Var_List$	$Var_List.type = Type.type$
$Type \rightarrow int$	$Type.type = entero$
$Type \rightarrow real$	$Type.type = real$
$Var_List_1 \rightarrow id, Var_List_2$	$id.type = Var_List_1.type$ $Var_List_2.type = Var_List_1.type$
$Var_List \rightarrow id$	$id.type = Var_List.type$

- ¿Qué atributos son heredados?
- ¿Qué atributos son sintetizados?
- Hacer el árbol de análisis con atributos de float x,y;

3. Traducción dirigida por la Sintaxis

Tipos de atributos



3. Traducción dirigida por la Sintaxis

Atrib. Sintetizados/ AS Ascendente

- Los algoritmos descritos en análisis sintáctico manipulan, en lugar de símbolos, estructuras que contienen la información semántica de cada símbolo: una estructura por símbolo; los campos de la estructura son los atributos del símbolo
- Los atributos sintetizados se pueden evaluar con un analizador sintáctico ascendente conforme la entrada es analizada.
- El analizador sintáctico conserva en su pila los valores de los atributos sintetizados asociados a los símbolos gramaticales.
- Cuando se hace una reducción se calculan los valores de los nuevos atributos sintetizados a partir de los atributos de la pila para los símbolos gramaticales del lado derecho de la producción

3. Traducción dirigida por la Sintaxis

Atrib. Sintetizados/ AS Ascendente

	ACCIONES
$L \rightarrow E$	$\{ \text{print}(E_1.\text{val}) \}$
$E \rightarrow E + T$	$\{ E_0.\text{val} = E_1.\text{val} + T_3.\text{val} \}$
$E \rightarrow T$	$\{ E_0.\text{val} = T_1.\text{val} \}$
$T \rightarrow T * F$	$\{ T_0.\text{val} = T_1.\text{val} * F_3.\text{val} \}$
$T \rightarrow F$	$\{ T_0.\text{val} = F_1.\text{val} \}$
$F \rightarrow (E)$	$\{ F_0.\text{val} = E_2.\text{val} \}$
$F \rightarrow \text{digito}$	$\{ F_0.\text{val} = \text{digito} \}$

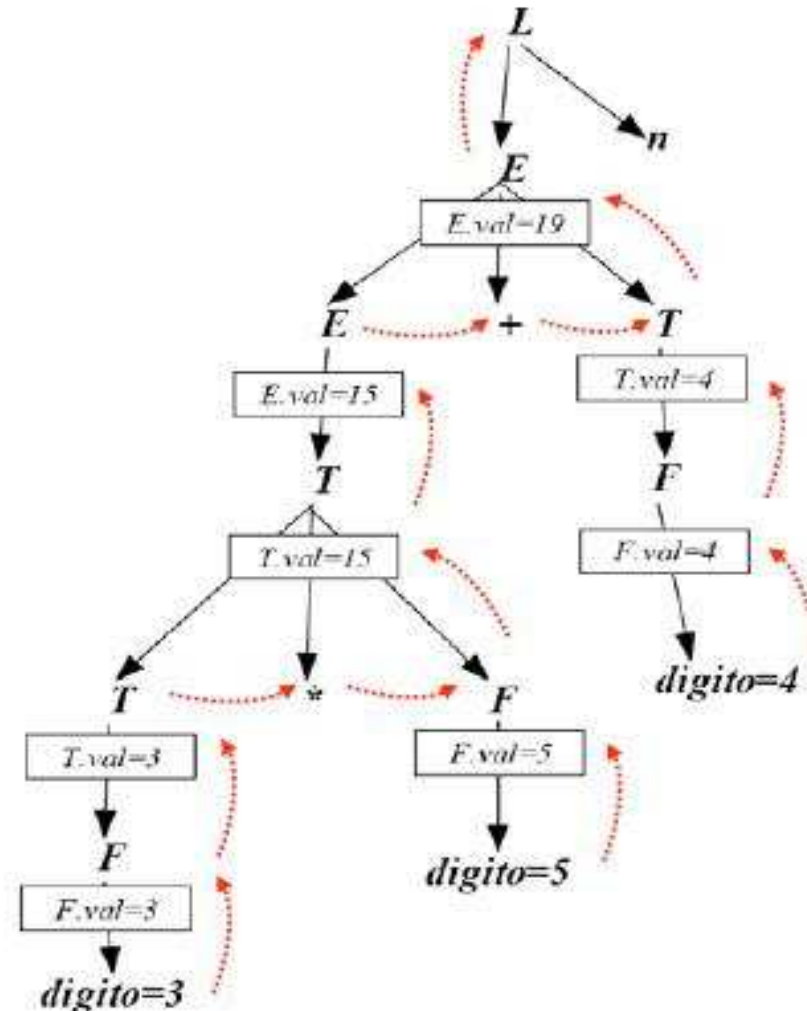
La evaluación de la expresión: $3 * 5 + 4n$

3. Traducción dirigida por la Sintaxis

Atrib. Sintetizados/ AS Ascendente

	ACCIONES
$L \rightarrow E$	{ <i>print</i> ($E_1.val$) }
$E \rightarrow E + T$	{ $E_0.val = E_1.val + T_3.val$ }
$E \rightarrow T$	{ $E_0.val = T_1.val$ }
$T \rightarrow T * F$	{ $T_0.val = T_1.val * F_3.val$ }
$T \rightarrow F$	{ $T_0.val = F_1.val$ }
$F \rightarrow (E)$	{ $F_0.val = E_2.val$ }
$F \rightarrow digito$	{ $F_0.val = digito$ }

La evaluación de la expresión: $3 * 5 + 4n$



3. Traducción dirigida por la Sintaxis

Evaluación en Gramáticas con Atributos Heredados

- La evaluación de un atributo heredado depende de los atributos asociados con los símbolos precedentes en la derivación.
- Requisitos: Realizar un análisis descendente.**
- Ejemplo: Dada la gramática siguiente:

Producciones

$D \rightarrow T L$

$T \rightarrow \text{int}$

$T \rightarrow \text{real}$

$L \rightarrow L , \text{id}$

$L \rightarrow \text{id}$

Reglas

$L_2.in = T_1.tipo;$

$T_0.tipo = \text{entero};$

$T_0.tipo = \text{real};$

$L_1.in = L_0.in;$

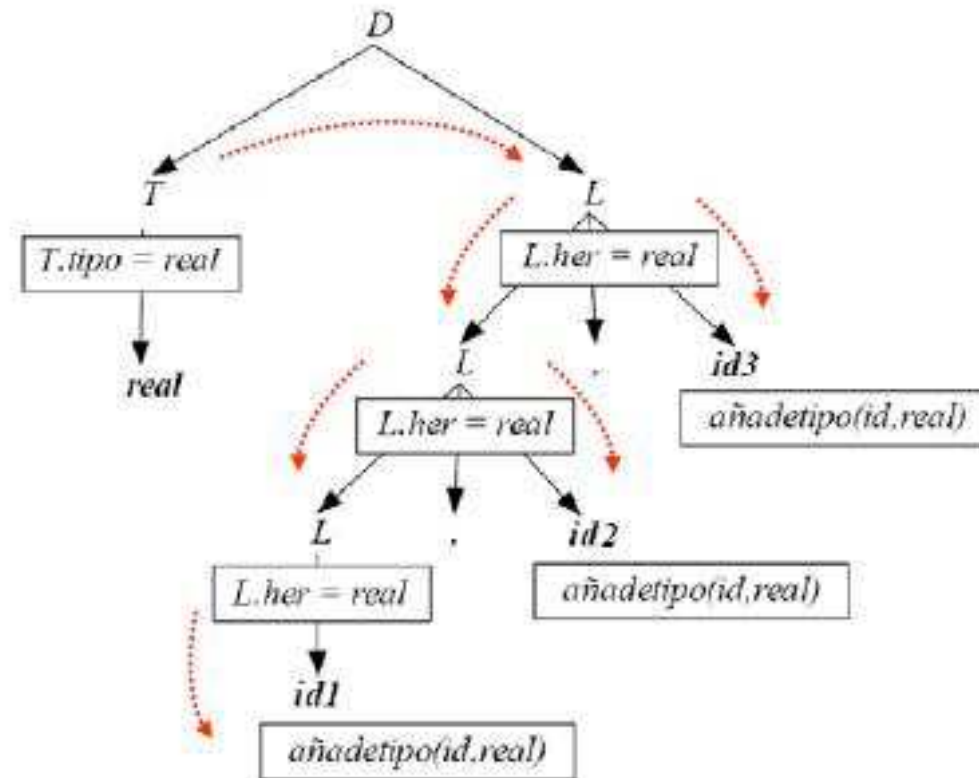
$\text{añadetipo}(\text{id}, L_0.in);$

$\text{añadetipo}(\text{id}, L_0.in);$

3. Traducción dirigida por la Sintaxis

Evaluación en Gramáticas con Atributos Heredados

- Y la cadena `Real id1, id2, id3`



3. Traducción dirigida por la Sintaxis

Evaluación de atributos

- ¿ Hasta qué punto los atributos se pueden calcular en el propio proceso de análisis sintáctico sin necesidad de hacer recorridos adicionales sobre el árbol? La respuesta depende del tipo de análisis que se utilice y del tipo de atributos definidos para los símbolos gramaticales.
- Los métodos LL y LR, al evaluar la cadena de componentes léxicos de izquierda a derecha, implican que no existan dependencias hacia atrás en el árbol (dependencias que apuntan desde la derecha hacia la izquierda).
- Las gramáticas de atributos que satisfacen esta condición se les llama **gramáticas de atributos-L** (gramática de atributos por la izquierda).

3. Traducción dirigida por la Sintaxis

Evaluación de atributos

- Si la **gramática es de atributos-L**, en la que además, los atributos **heredados no dependen de los sintetizados**, se pueden evaluar los atributos durante el proceso de **análisis sintáctico (descendente)**, considerando los atributos heredados como parámetros de los procedimientos y los atributos sintetizados como valores devueltos. De este modo, se evita almacenarlos en cada nodo.
- La función para un no_terminal A toma como argumentos un nodo y los valores de los atributos heredados para A y devuelve como resultados los valores de los atributos sintetizados para A.

3. Traducción dirigida por la Sintaxis

Evaluación de atributos

- Si los **atributos son sintetizados** se pueden evaluar con un **analizador sintáctico ascendente**. El analizador sintáctico puede mantener en la pila de análisis sintáctico los valores de los atributos asociados con los símbolos de la gramática. Cuando se hace una reducción se calcula el valor de los nuevos atributos a partir de los valores de los atributos de los símbolos de la parte derecha, que ya se encuentran en la pila.
- En el **resto de los casos**, hay que usar un **grafo de dependencias**

3. Traducción dirigida por la Sintaxis

Conclusión: cuándo se puede evaluar los atributos durante el análisis sintáctico

- Si los **atributos son sintetizados** se pueden evaluar con un **analizador sintáctico ascendente**
- Si los **atributos heredados no dependen de los sintetizados y la gramática es de atributos por la izda**, los atributos se pueden calcular durante el proceso de **análisis sintáctico descendente**, considerando los atributos heredados como parámetros de los procedimientos y los atributos sintetizados como valores devueltos

3. Traducción dirigida por la Sintaxis

Árbol con atributos (decorado)

- Los atributos se colocan al lado de los nodos del árbol de análisis sintáctico y se dibujan las dependencias entre ellos con flechas.
- Por ejemplo, el árbol de análisis con atributos de la expresión $(4-3)*2$ para la gramática de construcción de expresiones aritméticas simples:

3. Traducción dirigida por la Sintaxis

Orden de evaluación de un grafo de dependencias

- Un ordenamiento topológico de un grafo garantiza que las aristas van desde los nodos que aparecen primero en el ordenamiento a los que aparecen más tarde. Es decir: si $A \rightarrow B$, A aparecerá antes que B en el ordenamiento topológico.
- Un ordenamiento topológico da un orden válido para evaluar los atributos, es decir, recorriendo el grafo en ese orden se nos garantiza que tenemos los valores de los atributos ya calculados necesarios para evaluar las reglas semánticas en cada nodo que visitemos.

3. Traducción dirigida por la Sintaxis

Evaluación de atributos sintetizados

- Los valores de los atributos sintetizados se pueden calcular fácilmente mediante un recorrido ascendente (post-orden) del árbol sintáctico:

```
procedimiento EvaluarSintetizado(A:arbolSintactico){  
  Para cada hijo H de A hacer  
    EvaluarSintetizado(H);  
  Calcular atributos sintetizados de A  
}
```

3. Traducción dirigida por la Sintaxis

Evaluación de atributos sintetizados

- Ejemplo de **gramática S-Atribuida**:
 - Calculadora aritmética sencilla. Se desea evaluar expresiones a la vez que las analizamos. Sea el conjunto de producciones y acciones siguientes:

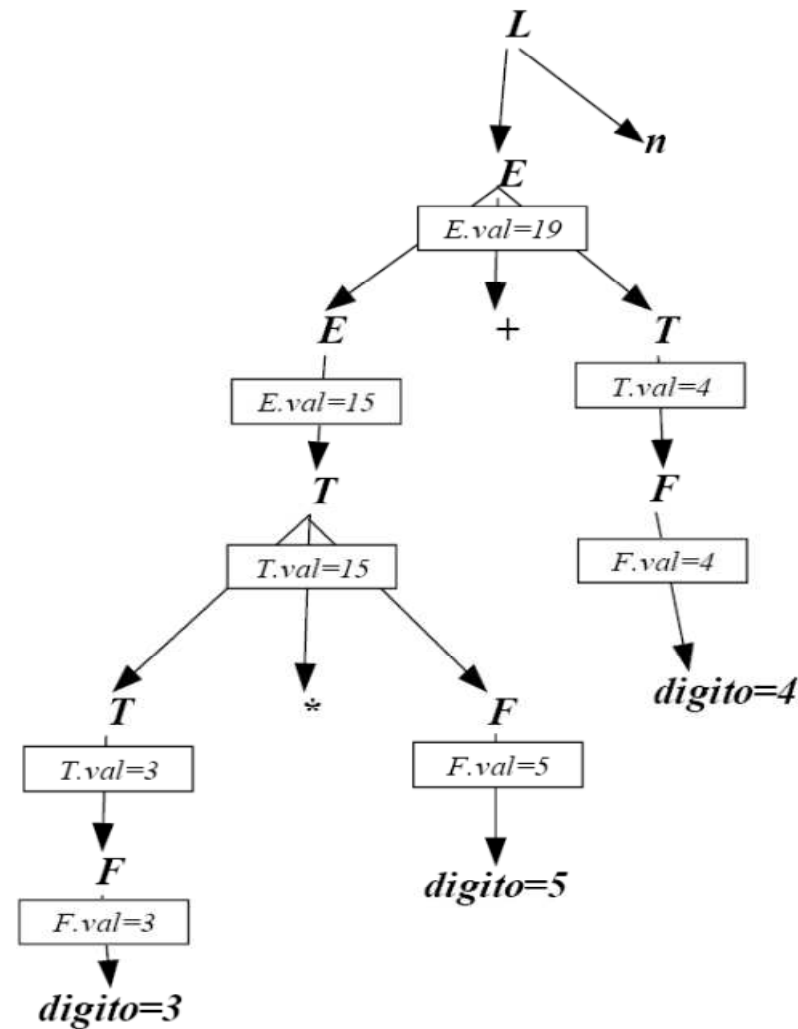
```
L → E n      { print (E1.val) } (* n = salto línea *)
E → E + T    { E0.val = E1.val + T3.val }
E → T        { E0.val = T1.val }
T → T * F    { T0.val = T1.val * F3.val }
T → F        { T0.val = F1.val }
F → (E)      { F0.val = E2.val }
F → digito   { F0.val = digito }
```

3. Traducción dirigida por la Sintaxis

Evaluación de atributos sintetizados

- Evaluación de la expresión "3 * 5 + 4"
- Resultado: se imprime el resultado de calcular 3 * 5 + 4.

$L \rightarrow E n$
 $E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid \text{digito}$



3. Traducción dirigida por la Sintaxis

Evaluación de atributos heredados

- Los valores de los atributos heredados se pueden calcular mediante un recorrido descendente (pre-orden) del árbol sintáctico:

```
procedimiento EvaluarHeredado(A:arbolSintactico){
  Para cada hijo H de A hacer{
    Calcular atributos heredados de H
    EvaluarHeredado(H);
  }
}
procedimiento EvaluarSintetizado(A:arbolSintactico){
  Para cada hijo H de A hacer
    EvaluarSintetizado(H);
  Calcular atributos sintetizados de A
}
```

3. Traducción dirigida por la Sintaxis

Evaluación de atributos heredados

- Ejemplo:

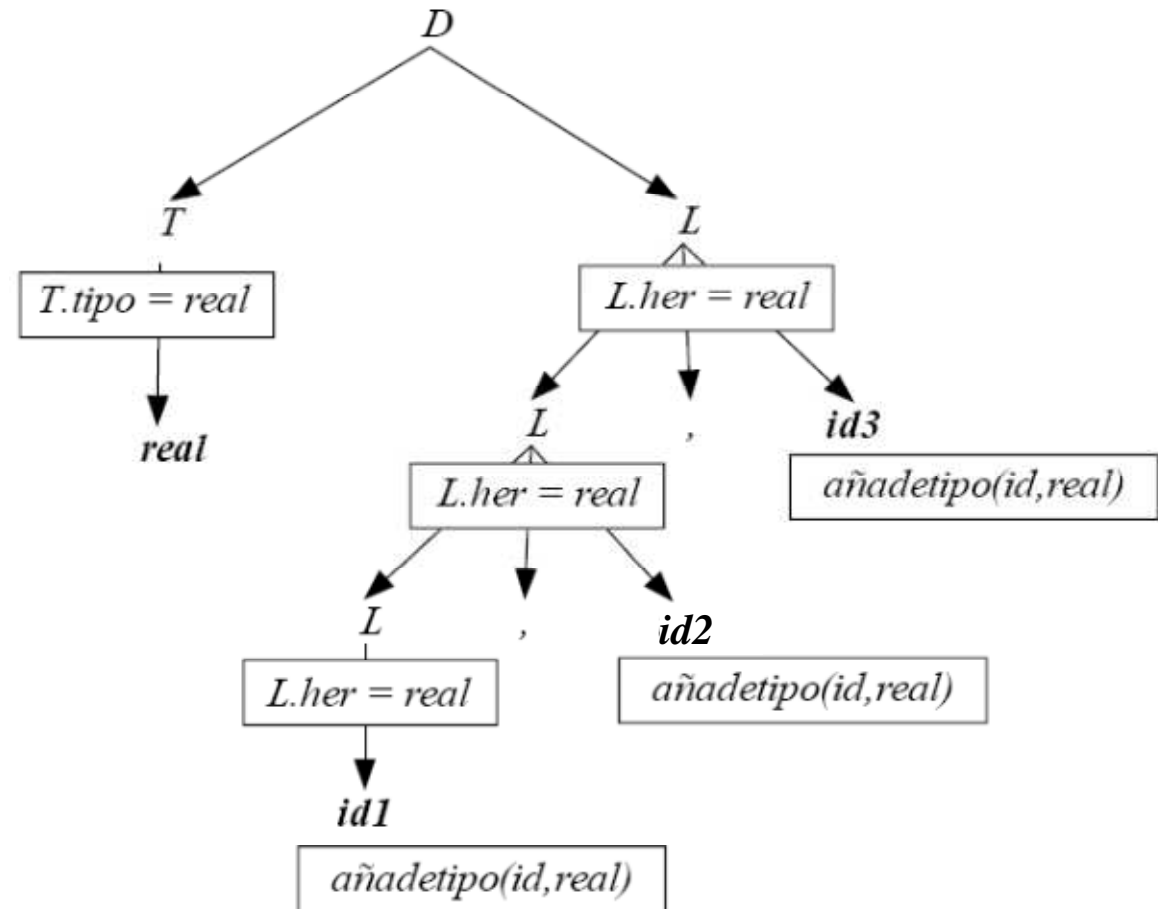
```
D → T L      { L.her = T.tipo; }
T → int      { T.tipo = entero; }
T → real     { T.tipo = real; }
L → L , id   { L1.her = L0.her; añadetipo(id,L0.her); }
L → id       { añadetipo(id, L.her); }
```


3. Traducción dirigida por la Sintaxis

Evaluación de atributos heredados

- Evaluar la expresión:
"real id1,id2,id3"

$D \rightarrow T L$
 $T \rightarrow \text{int}$
 $T \rightarrow \text{real}$
 $L \rightarrow L, \text{id}$
 $L \rightarrow \text{id}$



3. Traducción dirigida por la Sintaxis

Evaluación de atributos

- El cálculo de los atributos depende de la estructura de la gramática.
- Es posible simplificar el cálculo mediante una modificación de las reglas gramaticales.
- **Teorema de Knuth:** Dada una gramática con atributos, todos los atributos heredados se pueden convertir en sintetizados modificando adecuadamente la gramática, sin cambiar el lenguaje.
 - En la práctica no se utiliza demasiado, pues puede generar gramáticas y reglas semánticas más complejas que las originales.

3. Traducción dirigida por la Sintaxis

Métodos para la evaluación de los atributos

- **Métodos basados en grafos de dependencias:** en el momento de la compilación estos métodos obtienen un orden de evaluación a partir del grafo de dependencias sobre el árbol de análisis sintáctico para la entrada dada (el programa fuente). Si hay ciclos no funciona.
- **Métodos basados en reglas:** en el momento de la construcción del compilador, para cada producción queda predeterminado por el diseñador del compilador el orden de evaluación de los atributos de esa construcción lingüística, y así la forma de recorrer el árbol para calcular ese atributo

3. Traducción dirigida por la Sintaxis

Evaluación de atributos: grafos de dependencias

- Motivaciones
 - ¿Cuándo debe ejecutarse una acción semántica?
 - Los atributos no pueden calcularse en cualquier orden
 - Todos los atributos de los que depende deben tener valor

- Grafo de dependencias
 - Determina el flujo de información
 - Determina un orden de asignación de valores
 - Induce cierto orden de ejecución de las reglas

3. Traducción dirigida por la Sintaxis

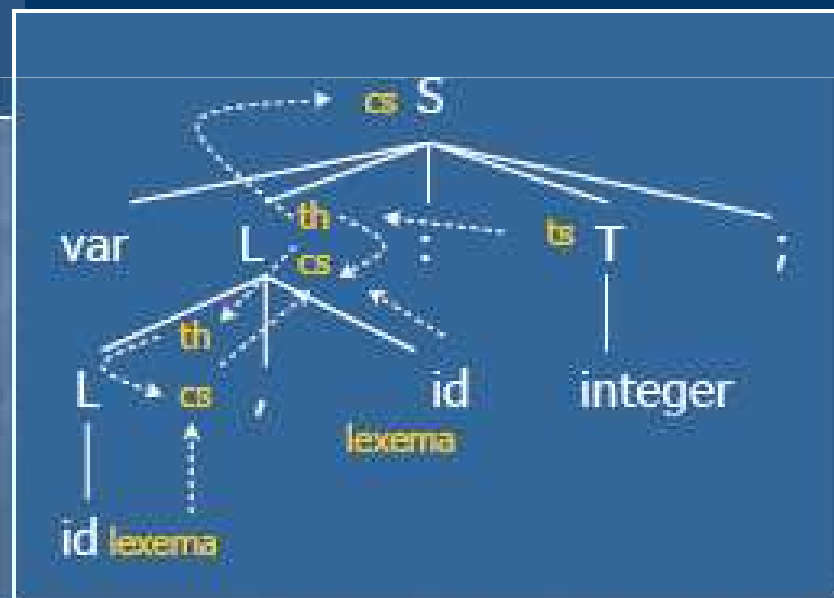
Evaluación de atributos: grafos de dependencias

- Cuando aparecen definidos atributos sintetizados y heredados, es necesario establecer un ORDEN DE EVALUACIÓN DE LOS ATRIBUTOS mediante un GRAFO de DEPENDENCIA,
- El grafo tiene un nodo por cada atributo, y una flecha de b a c si c depende de b . Se construye con el algoritmo siguiente:
- **Nodos:**
 - Para cada nodo n del árbol de análisis hacer:
 - Para cada atributo a del símbolo del nodo n hacer:
 - Construir un nodo etiquetado con a .
- **Arcos:**
 - Para cada nodo n del árbol de análisis hacer:
 - Para cada regla semántica $b=f(e_1, \dots, e_k)$ asociada con la producción usada en el nodo n hacer:
 - Para $i=1, \dots, k$ hacer:
 - Trazar arcos desde e_i a b .

3. Traducción dirigida por la Sintaxis

Evaluación de atributos: grafos de dependencias

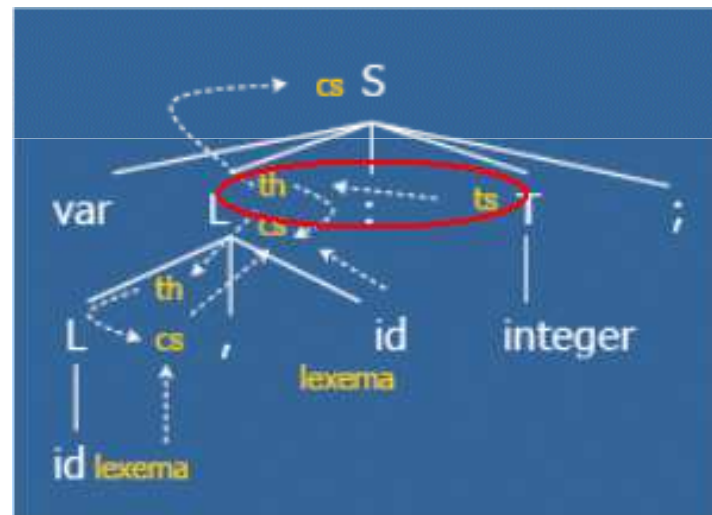
$S ::= \text{var } L : T ;$	$L.th := T.ts$
	$S.cs := L.cs$
$L ::= L_1, \text{id}$	$L.cs := L_1.cs \parallel L.th \parallel \text{id.lexema} \parallel ";"$
	$L_1.th := L.th$
$L ::= \text{id}$	$L.cs := L.th \parallel \text{id.lexema} \parallel ";"$
$T ::= \text{real}$	$T.ts := \text{float}$
$T ::= \text{integer}$	$T.ts := \text{int}$



3. Traducción dirigida por la Sintaxis

Evaluación de atributos: grafos de dependencias

- Para construir traductores de una sola pasada...
 - Los atributos no pueden depender de otros calculados a su derecha:



3. Traducción dirigida por la Sintaxis

Evaluación de atributos: Compatibilidad

- Los dos órdenes (derivación y cálculo de atributos) pueden no ser compatibles
- En compiladores de 2 ó más pasos no es necesario que lo sean:
 - Primera pasada – generación del árbol
 - Segunda pasada – anotación del árbol
- En analizadores ascendentes los dos órdenes son compatibles si en la gramática todos los atributos son sintetizados (su valor se calcula utilizando los atributos de los símbolos de la parte derecha)
- En los descendentes, si los atributos heredados no dependen de los sintetizados y la gramática es de atributos por la izda.

3. Traducción dirigida por la Sintaxis

Técnica general del análisis semántico en compiladores de 2 o más pasos

1. Construir el árbol del derivación.
2. Determinar las dependencias entre los atributos mediante el estudio de las acciones semánticas de la gramática.
3. Determinar un orden entre los atributos del árbol, compatible con las dependencias obtenidas en el paso anterior.
4. Establecer un recorrido del árbol compatible con el orden del paso 3.
5. Ejecutar las acciones semánticas que aparecen al recorrer el árbol según indica el paso 4.

3. Traducción dirigida por la Sintaxis

Traducción descendente

- En esta sección, se implantarán las definiciones con atributos por la izquierda durante el análisis sintáctico predictivo (descendente).
- Se trabaja con esquemas de traducción, así que puede ser explícito en cuanto al orden en que tienen lugar las acciones y las evaluaciones de los atributos.
- También se amplía el algoritmo para la eliminación de la recursión por la izquierda a esquemas de traducción con atributos sintetizados.

3. Traducción dirigida por la Sintaxis

Eliminación de la recursión por la izquierda de un esquema de traducción

- Motivación: Como la mayoría de los operadores aritméticos son asociativos por la izquierda, es natural utilizar gramáticas recursivas por la izquierda para las expresiones.
- Ahora se amplía el algoritmo de eliminación de recursividad por la izquierda, visto durante el tema 2, para admitir atributos cuando se transforma la gramática subyacente de un esquema de traducción.
- La transformación se aplica a esquemas de traducción con atributos sintetizados.
- Ello permite que muchas de las definiciones vistas hasta ahora, se implanten mediante un análisis sintáctico predictivo.

3. Traducción dirigida por la Sintaxis

Eliminación de la recursión por la izquierda de un esquema de traducción

- Ejemplo: Transformación de gramática para exp. arit.

$$E \rightarrow E_1 + T \quad \{E.val := E_1.val + T.val\}$$

$$E \rightarrow E_1 - T \quad \{E.val := E_1.val - T.val\}$$

$$E \rightarrow T \quad \{E.val := T.val\}$$

$$T \rightarrow (E) \quad \{T.val := E.val\}$$

$$T \rightarrow num \quad \{T.val := num.val\}$$



$$E \rightarrow T \quad \{R.h := T.val\}$$

$$R \quad \{E.val := R.s\}$$

$$R \rightarrow +$$

$$T \quad \{R_1.h := R.h + T.val\}$$

$$R_1 \quad \{R.s := R_1.s\}$$

$$R \rightarrow -$$

$$T \quad \{R_1.h := R.h - T.val\}$$

$$R_1 \quad \{R.s := R_1.s\}$$

$$R \rightarrow \in \quad \{R.s := R.h\}$$

$$T \rightarrow ($$

E

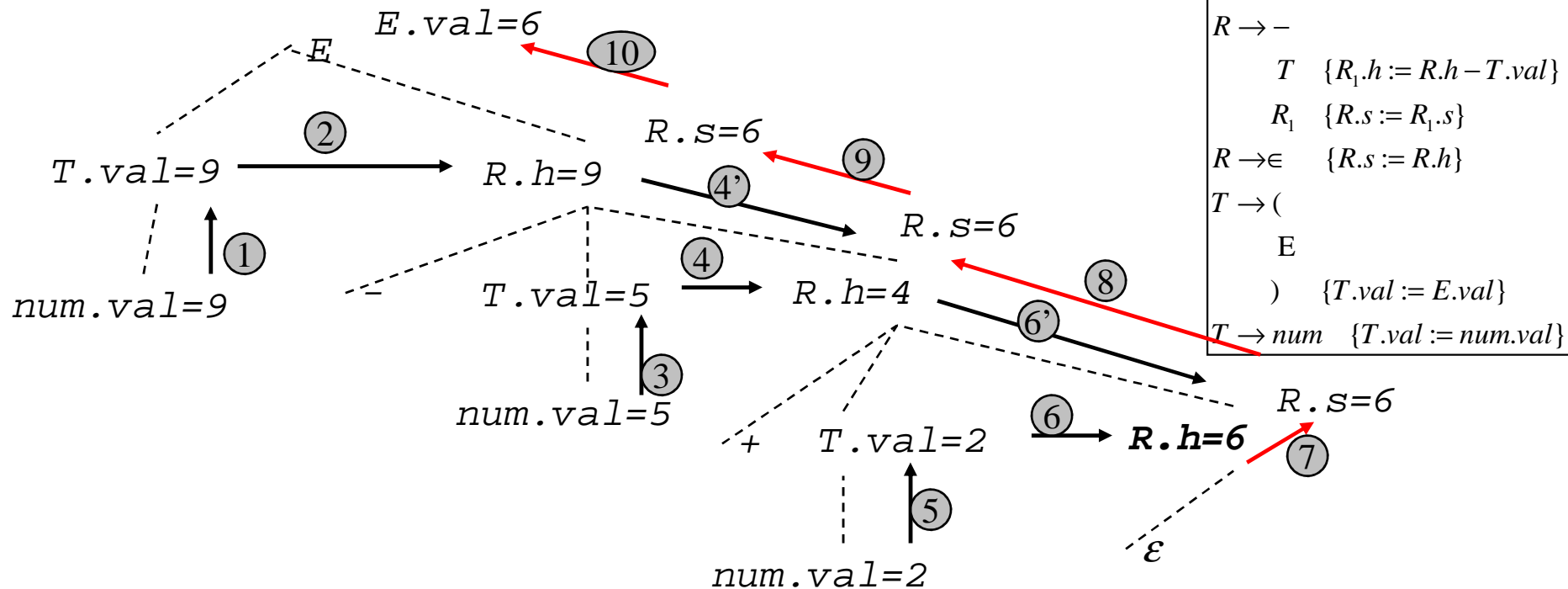
$$) \quad \{T.val := E.val\}$$

$$T \rightarrow num \quad \{T.val := num.val\}$$

3. Traducción dirigida por la Sintaxis

Eliminación de la recursión por la izquierda de un esquema de traducción

- Ejemplo: 9-5+2



3. Traducción dirigida por la Sintaxis

Algoritmo de eliminación de la recursión por la izquierda de un esquema de traducción

- Supóngase que se tiene el siguiente esquema de traducción:

$$\begin{array}{l}
 A \rightarrow A_1 Y \\
 A \rightarrow X
 \end{array}
 \xrightarrow{\text{Eliminar rec. izda}}
 \begin{array}{l}
 A \rightarrow X R \\
 R \rightarrow Y R_1 \\
 R \rightarrow \epsilon
 \end{array}$$

$$\begin{array}{l}
 A \rightarrow A_1 Y \quad \{A.a := g(A_1.a, Y.y)\} \\
 A \rightarrow X \quad \{A.a := f(X.x)\}
 \end{array}
 \xrightarrow[\text{con esq. traduc.}]{\text{Eliminar rec. izda}}
 \begin{array}{l}
 A \rightarrow X \quad \{R.h := f(X.x)\} \\
 R \quad \{A.a := R.s\} \\
 R \rightarrow Y \quad \{R_1.h := g(R.h, Y.y)\} \\
 R_1 \quad \{R.s := R_1.s\} \\
 R \rightarrow \epsilon \quad \{R.s := R.h\}
 \end{array}$$

3. Traducción dirigida por la Sintaxis

Construcción de un traductor predictivo dirigido por la sintaxis

Entrada: Un esquema de traducción dirigido por la sintaxis con una gramática subyacente para el análisis predictivo.

Salida: Código del traductor dirigido por la sintaxis.

Método: La técnica es una modificación de la construcción de analizadores sintácticos predictivos de la sección 2.4.

1. Para cada no terminal A , constrúyase una función que tenga un parámetro formal por cada atributo heredado de A y que devuelva los valores de los atributos sintetizados de A . Para simplificar, se supone que cada no terminal sólo tiene un atributo sintetizado. La función de A tiene una variable local para cada atributo de cada símbolo gramatical que aparezca en una producción para A .
2. Al igual que en el algoritmo de reconocimiento predictivo, el código para el no terminal A decide qué producción utilizar basándose en el símbolo en curso de entrada.

3. Traducción dirigida por la Sintaxis

Construcción de un traductor predictivo dirigido por la sintaxis (II)

3. El código asociado con cada producción hace lo siguiente. Se consideran los componentes léxicos, no terminales y acciones del lado derecho de la producción de izquierda a derecha.
 - i) Para el componente léxico X con atributo sintetizado x , guárdese el valor de x en la variable declarada $X.x$. Después genérese una llamada para concordar el componente léxico X y aváncese en la entrada.
 - ii) Para el no terminal B , genérese una asignación $c := B(b_1, b_2, \dots, b_k)$ con la llamada a una función en el lado derecho, donde b_1, \dots, b_k son las variables para los atributos heredados de B y c es la variable del atributo sintetizado de B .

3. Traducción dirigida por la Sintaxis

Construcción de un traductor predictivo dirigido por la sintaxis (III)

iii) Para el caso de una acción, cópiese el código dentro del analizador sintáctico, sustituyendo cada referencia a un atributo por la variable correspondiente a dicho atributo.