

Procesadores de lenguajes. Práctica 7.

ANALIZADORES SINTÁCTICOS ASCENDENTES

A solid orange horizontal bar at the bottom of the slide.

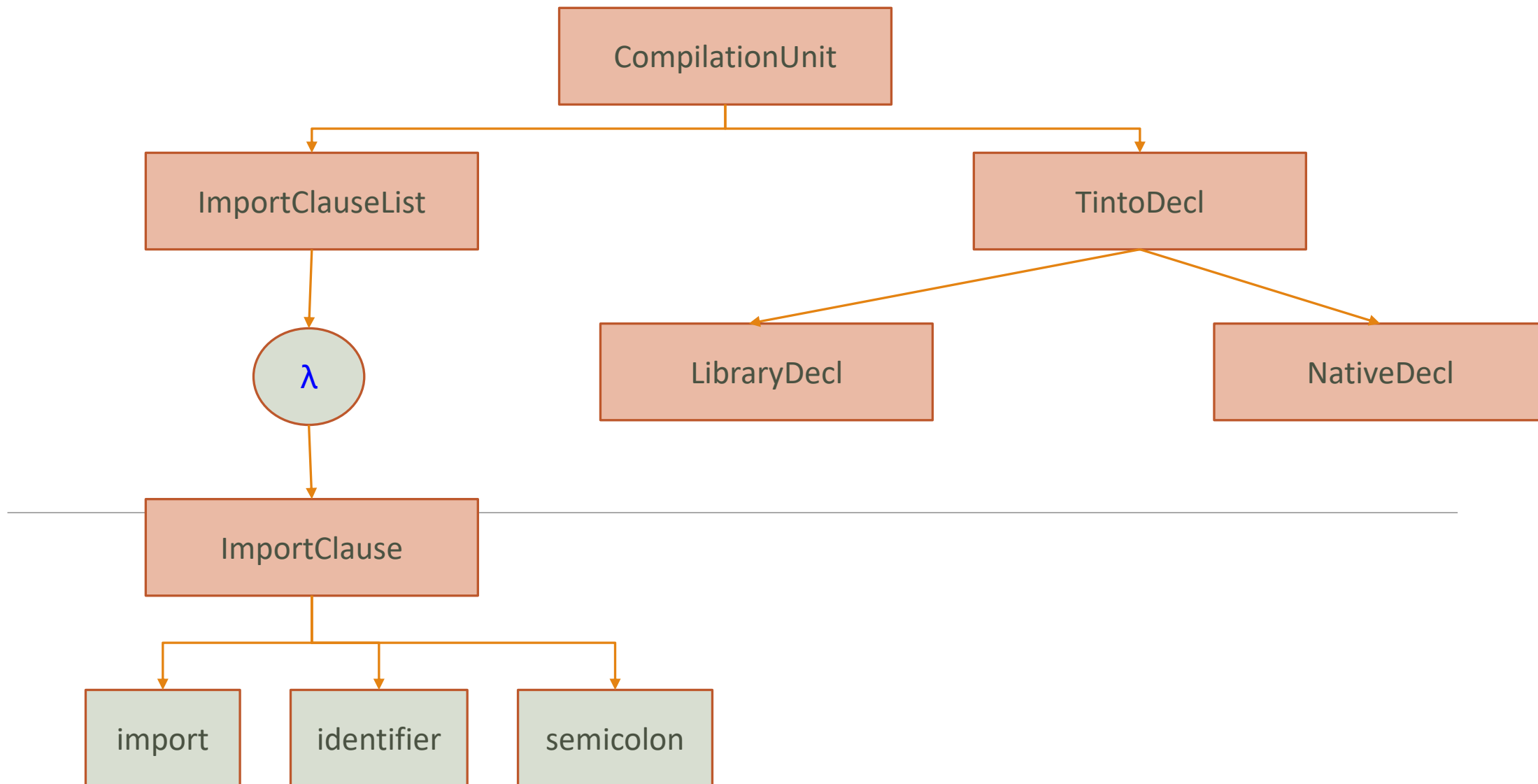
Índice

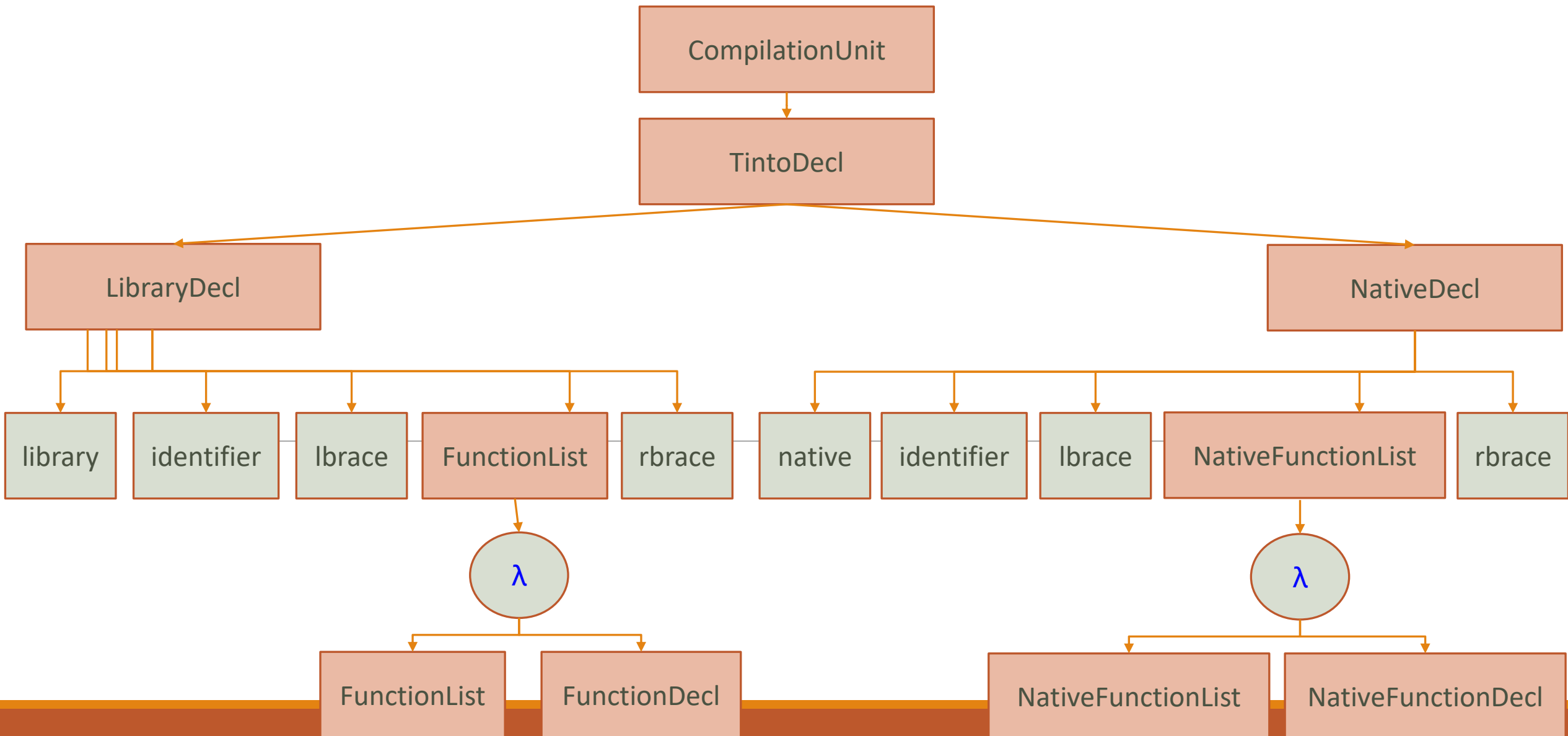
1.1 Esquemas Tinto

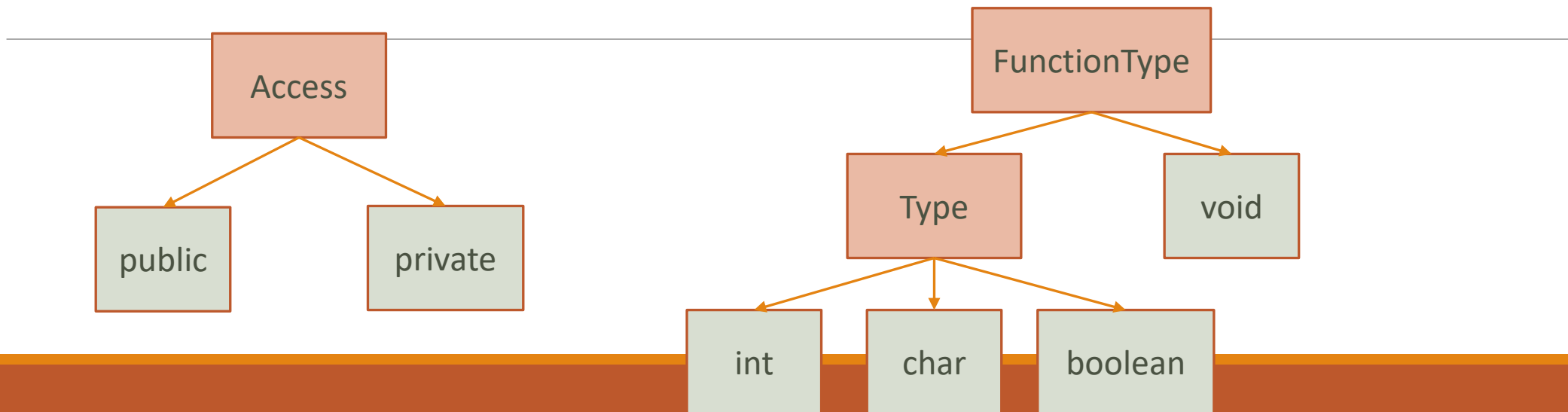
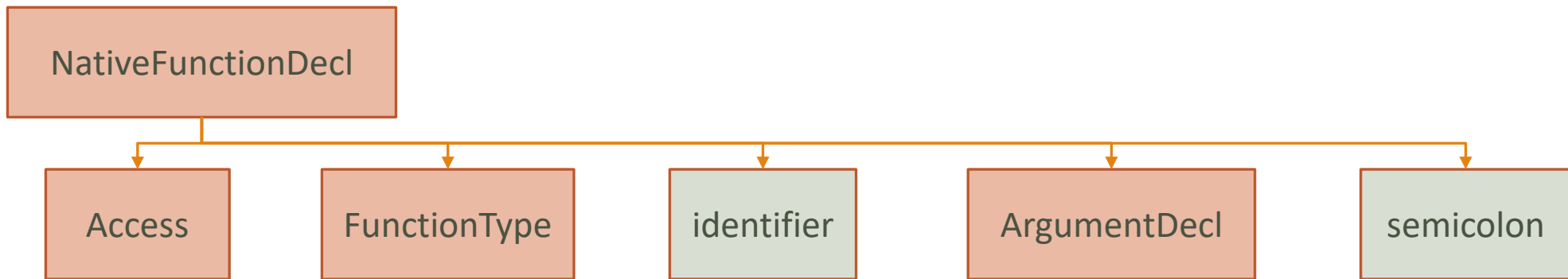
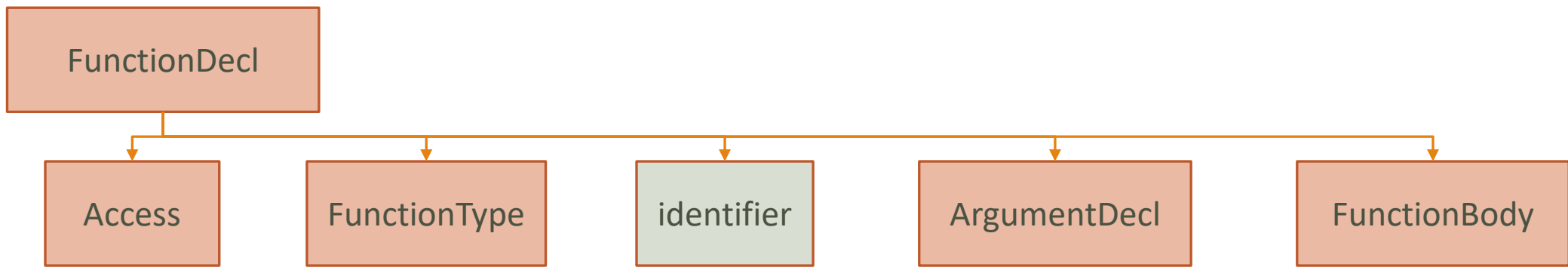
1.2 Descripción del método

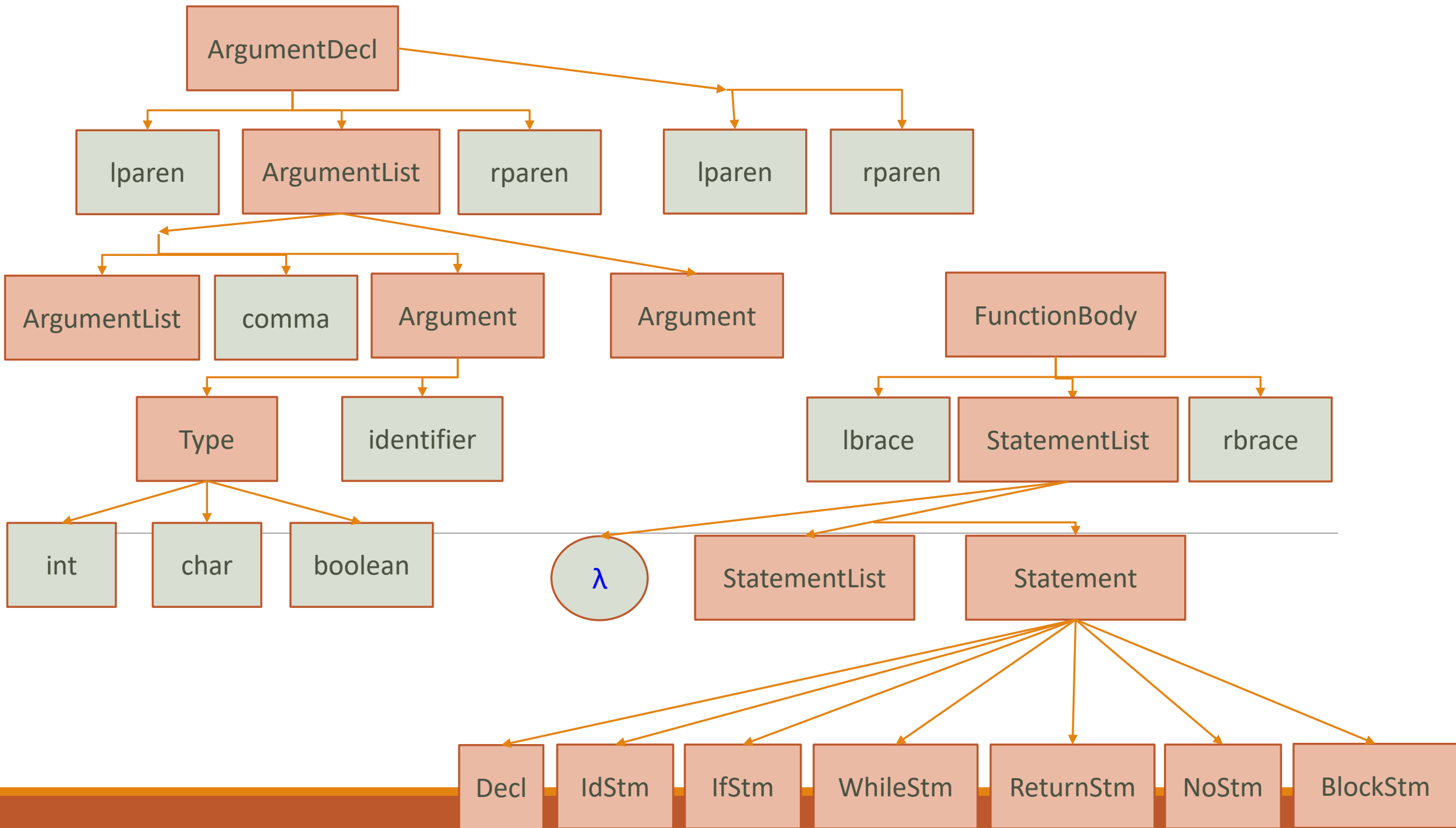
1.3 Ejemplo clase

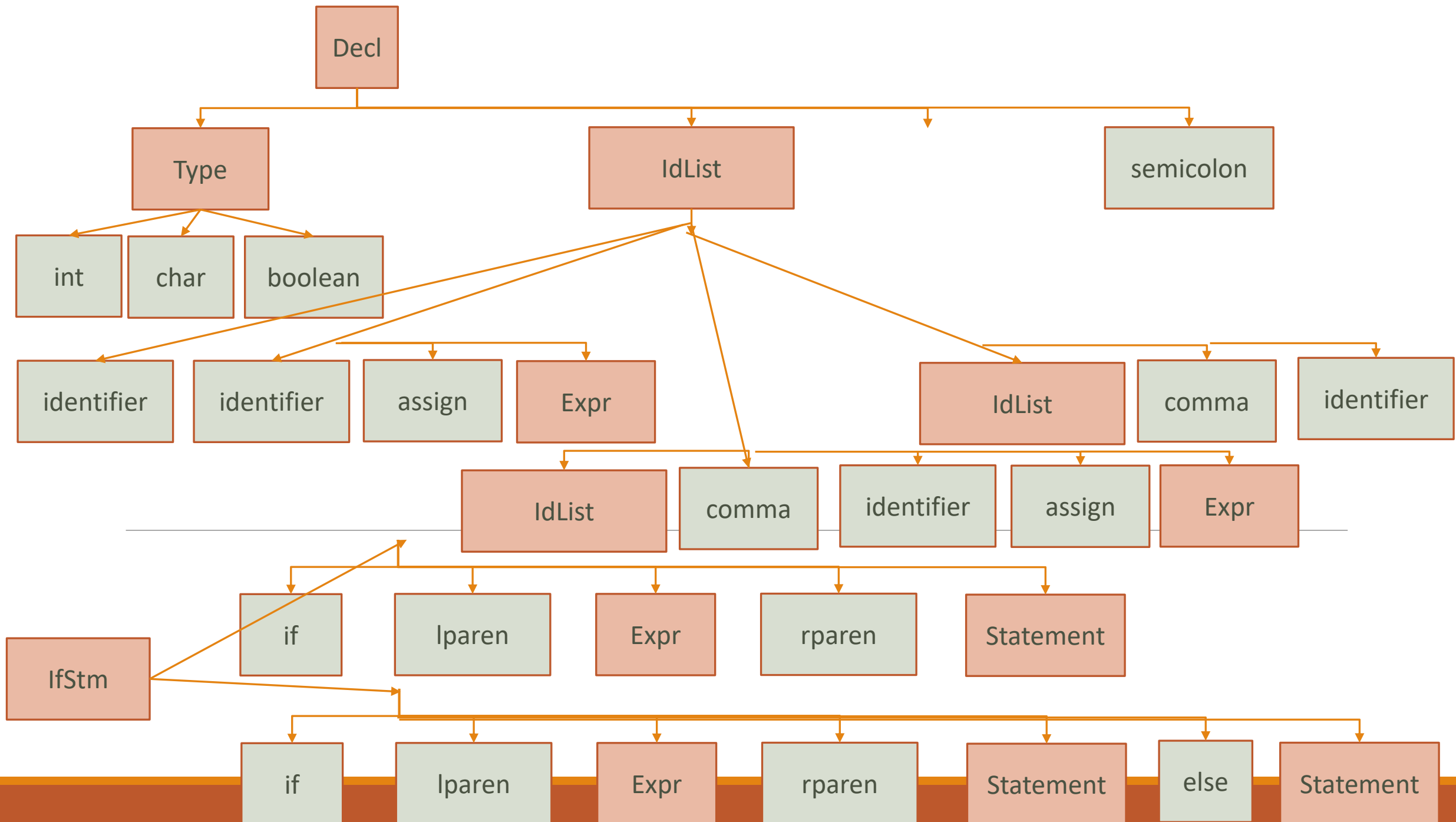
1.4 Implementación en el trabajo final

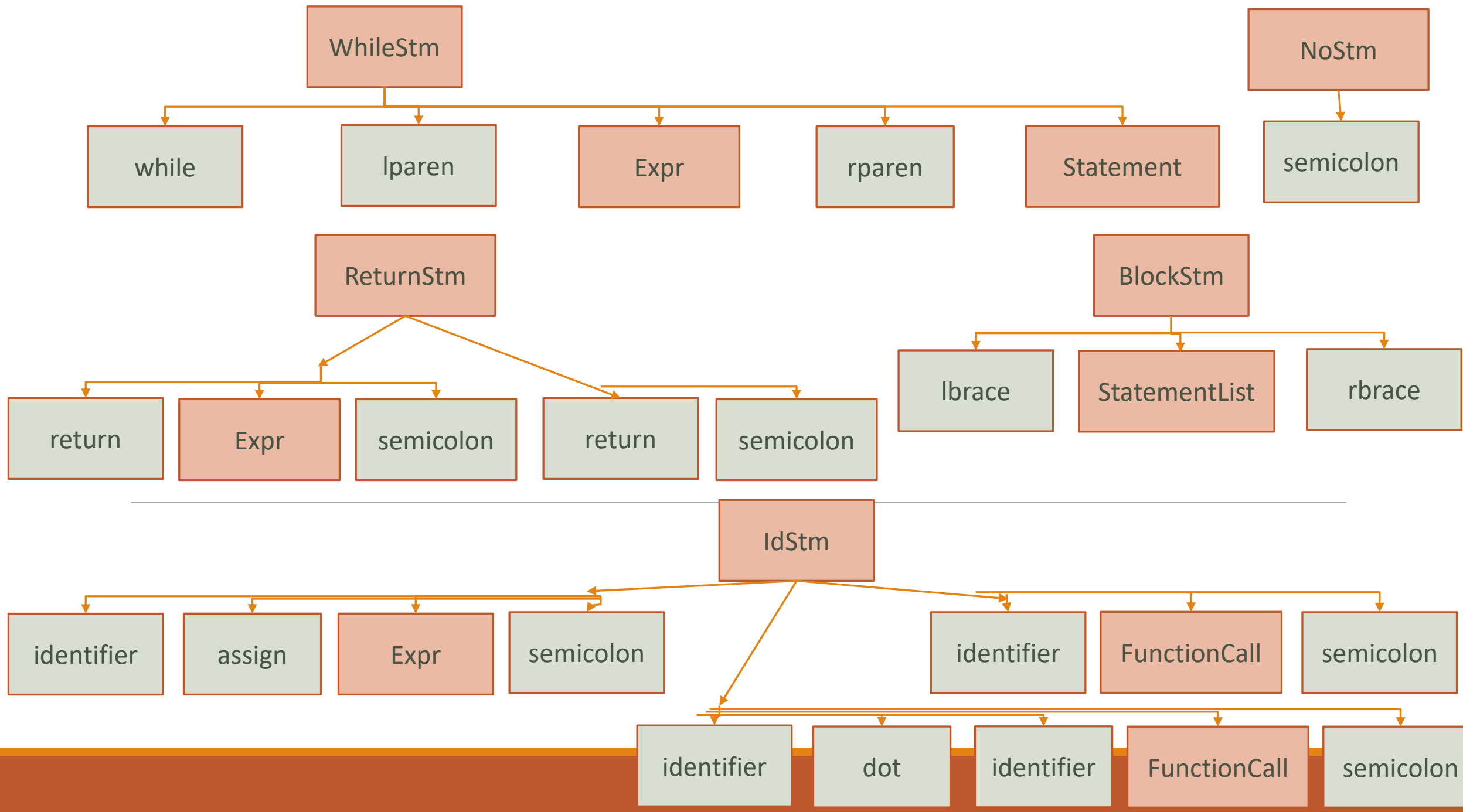


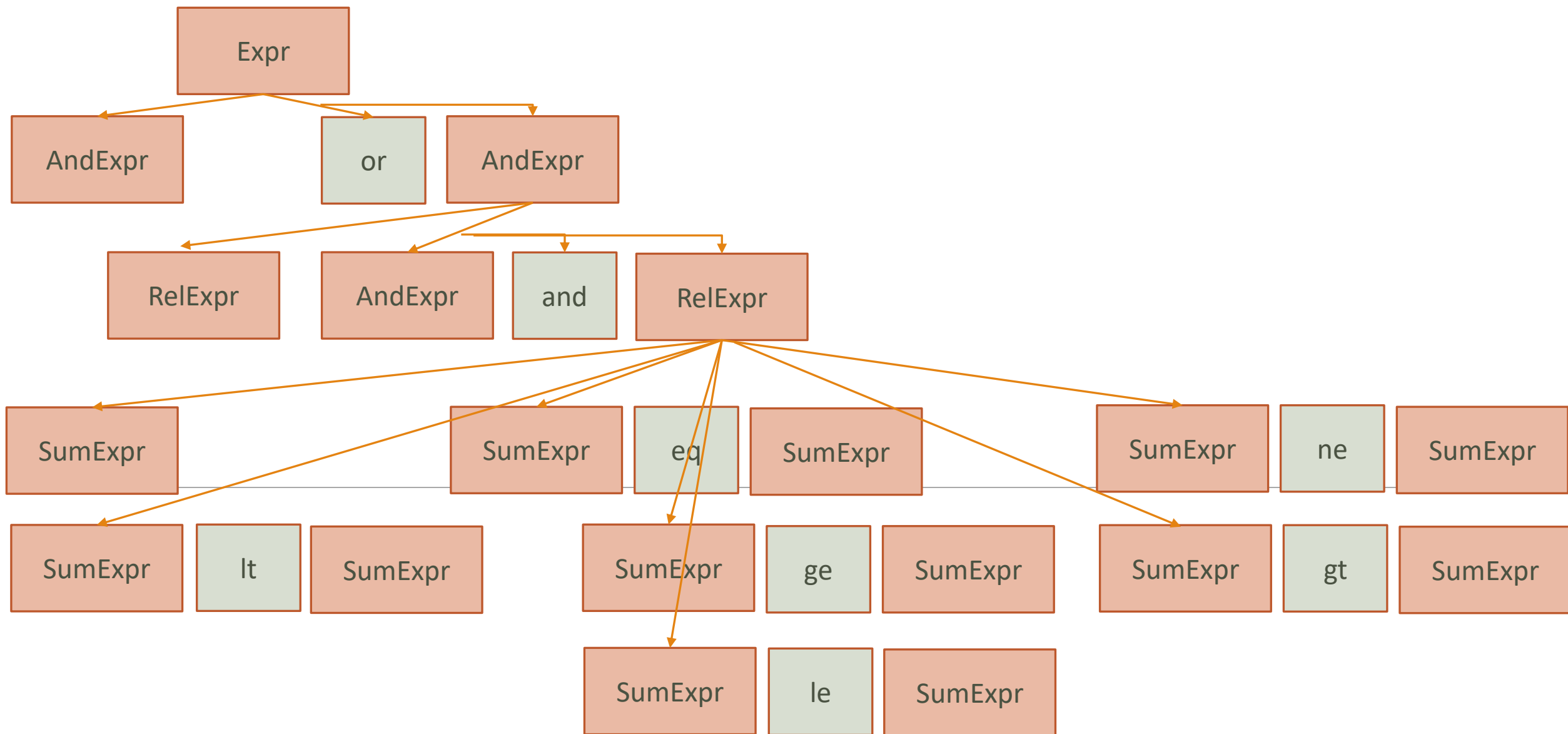


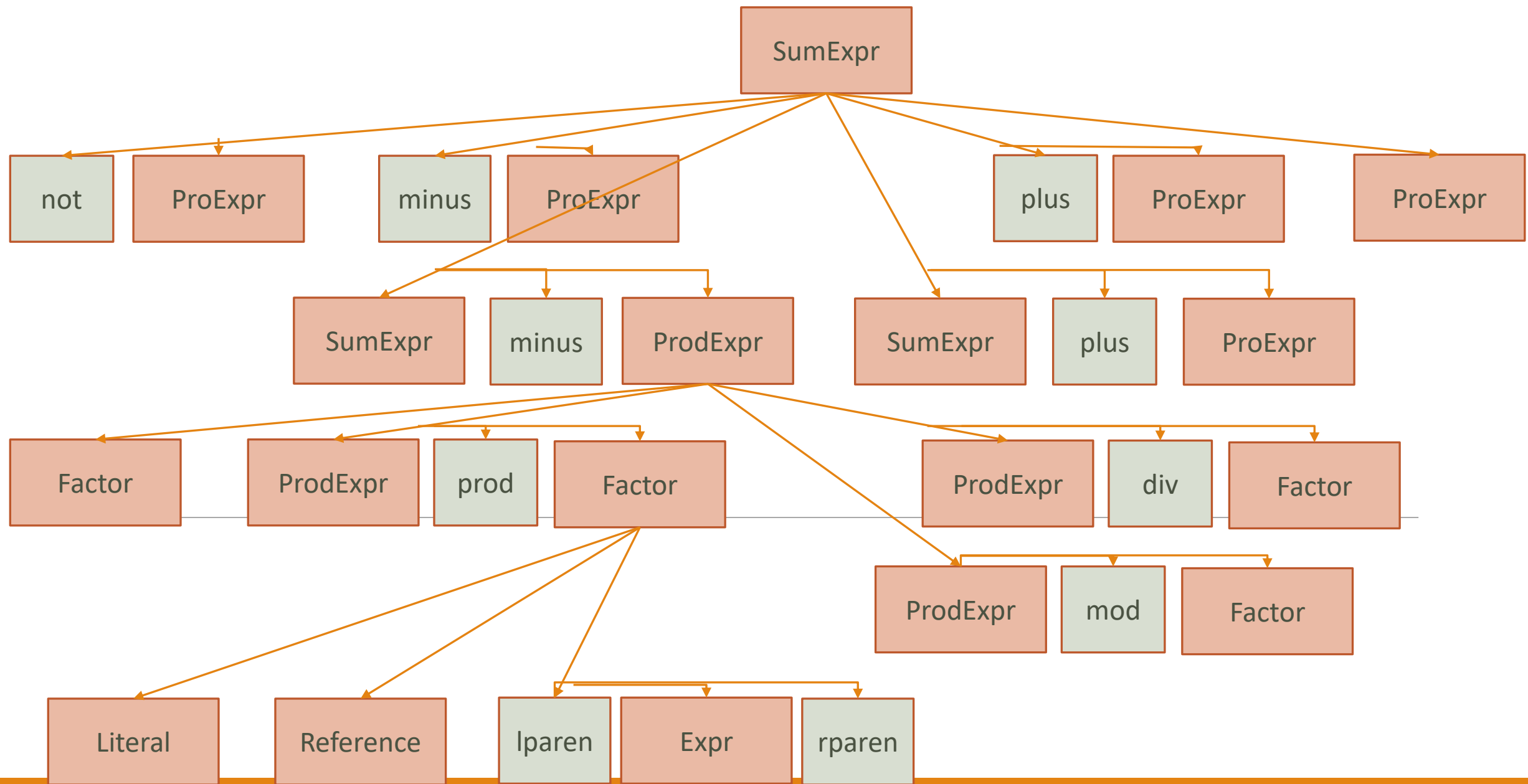


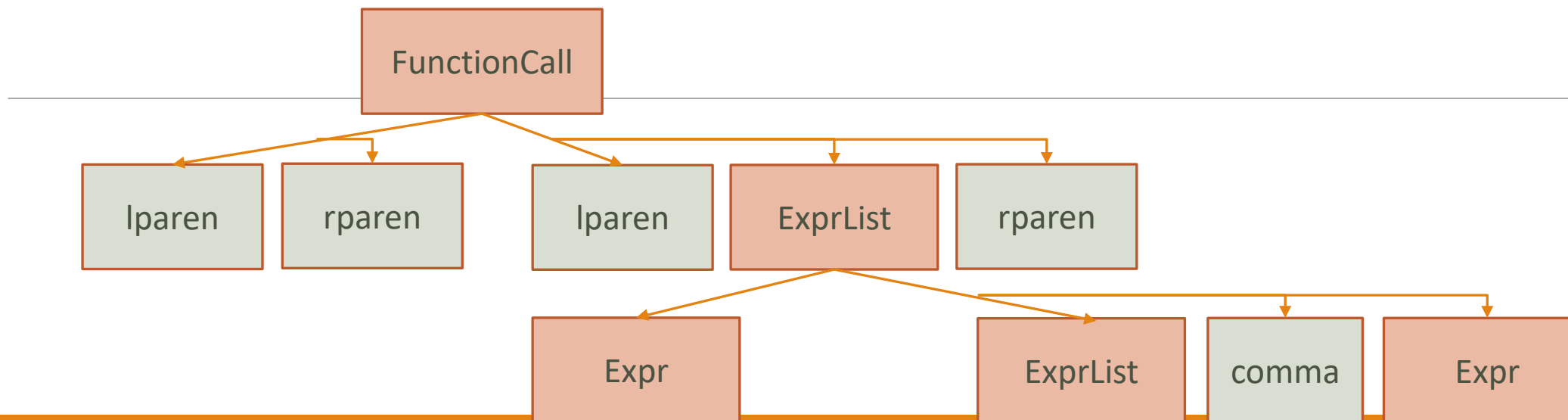
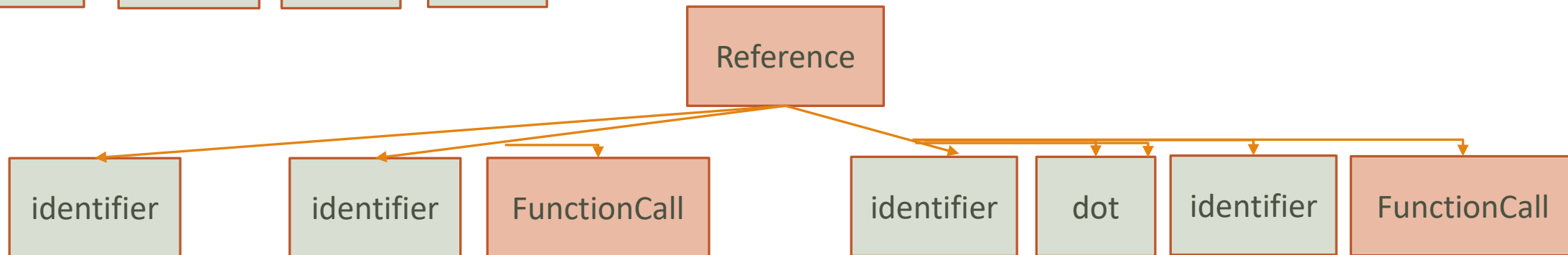
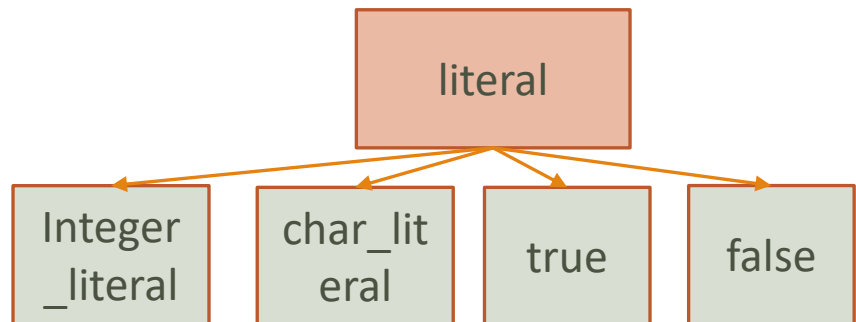












1.2 Descripción del método

- ***step()***: realiza una iteración del algoritmo de desplazamiento/reducción, es decir, lee el estado de la cima de la pila, considera el siguiente token y busca la celda correspondiente en la tabla de acción. Si la celda es nula lanza un error sintáctico. Si la celda es una acción de desplazamiento ejecuta el método *shiftAction()* y devuelve *false* (para que el bucle de análisis continúe su ejecución). Si la celda contiene una acción de reducción ejecuta el método *reduceAction()* y devuelve *false* (para que el bucle de análisis continúe su ejecución). Si la celda contiene una acción de aceptar devuelve *true* para que finalice el bucle de análisis.
- ***shiftAction()***: realiza una acción de desplazamiento, es decir, lee el siguiente token del analizador léxico y apila el estado correspondiente.
- ***reduceAction()***: realiza una acción de reducción, es decir: busca la regla a reducir; desapila tantos estados como símbolos tenga la regla; estudia el estado de la cima de la pila; busca en la tabla IrA la celda correspondiente al estado de la cima y al símbolo no terminal de la regla; y apila el valor de la celda de la tabla IrA.

1.3 Ejemplo clase

Sea la gramática:

- $\langle S \rangle \rightarrow \langle B \rangle \langle A \rangle \text{ end}$
- $\langle A \rangle \rightarrow \text{begin } \langle C \rangle$
 - $\langle C \rangle \rightarrow \text{codigo}$
 - $\langle B \rangle \rightarrow \text{tipo}$
 - $\langle B \rangle \rightarrow \text{id } \langle B \rangle$

Init_rules:

- * R0 ... X ::= CompilationUnit
- * R1 ... CompilationUnit ::= $\langle B \rangle \langle A \rangle \text{ end}$
- * R2 ... $\langle A \rangle ::= \text{begin } \langle C \rangle$
- * R3 ... $\langle C \rangle ::= \text{codigo}$
- * R4 ... $\langle B \rangle ::= \text{tipo}$
- * R5 ... $\langle B \rangle ::= \text{id } \langle B \rangle$

1: <S> -> <A> end
 2: <A> -> begin <C>
 3: <C> -> código
 4: -> tipo
 5: -> id

d salta al estado
 r reduce la regla

Estado		
0	X -> * <S> <S> -> * <A> end -> * tipo -> * id 	<S> -> 1 -> 2 tipo -> 3 id -> 4
1	X -> <S> *	
2	<S> -> * <A> end <A> -> * begin <C>	<A> -> 5 begin -> 6
3	 -> tipo *	
4	 -> id * -> * tipo -> * id 	tipo -> 3 id -> 4 -> 7
5	<S> -> <A> * end	end -> 8
6	<A> -> begin * <C> <C> -> * código	<C> -> 9 código -> 10
7	 -> id *	
8	<S> -> <A> end *	
9	<A> -> begin <C> *	
10	<C> -> código *	

1.3 Ejemplo clase pag 24 tema 4

Estado	end	begin	codigo	tipo	id	\$	S	A	B	C
0				d3	d4		1		2	
1						aceptar				
2		d6						5		
3	r4	r4	r4	r4	r4	r4				
4				d3	d4				7	
5	d8									
6			d10							9
7	r5	r5	r5	r5	r5	r5				
8	r1	r1	r1	r1	r1	r1				
9	r2	r2	r2	r2	r2	r2				
10	r3	r3	r3	r3	r3	r3				

