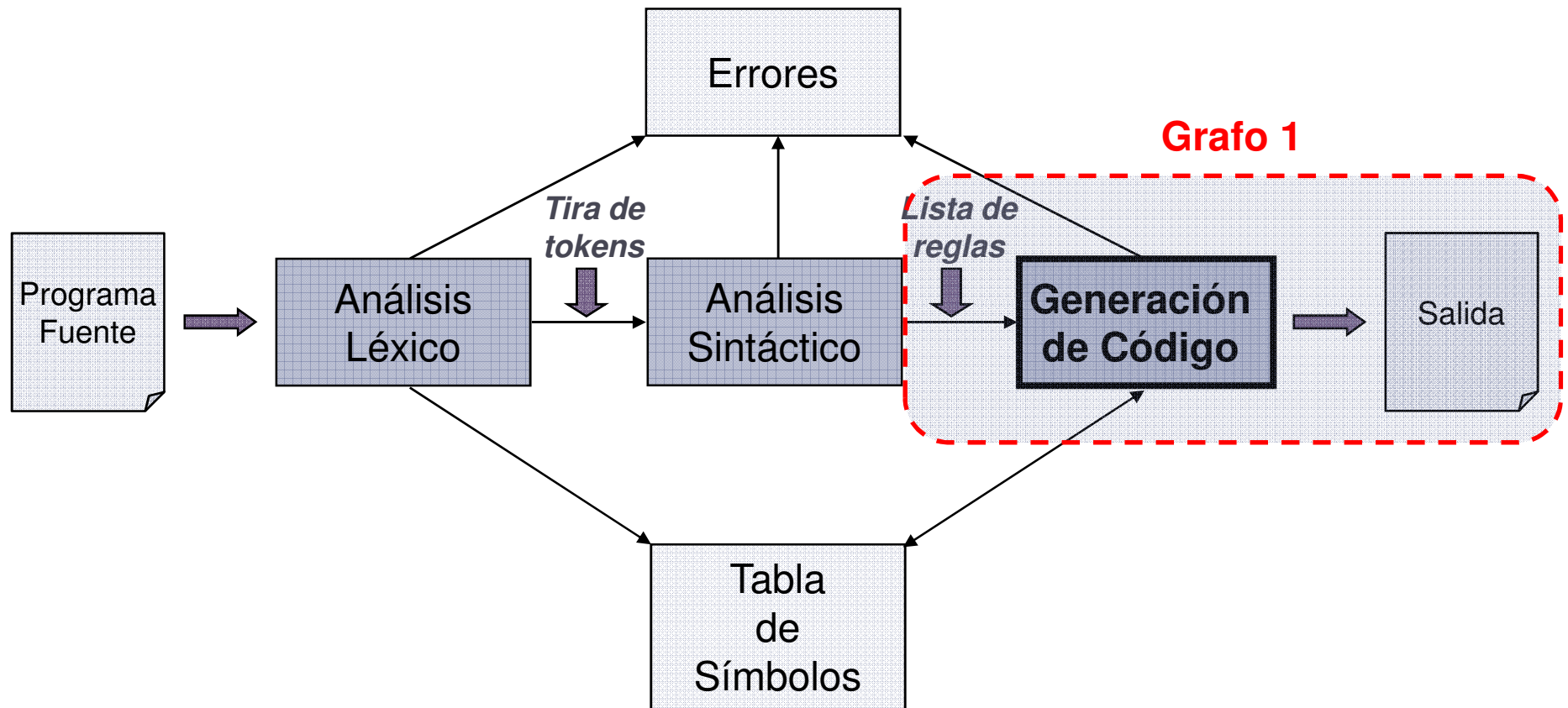


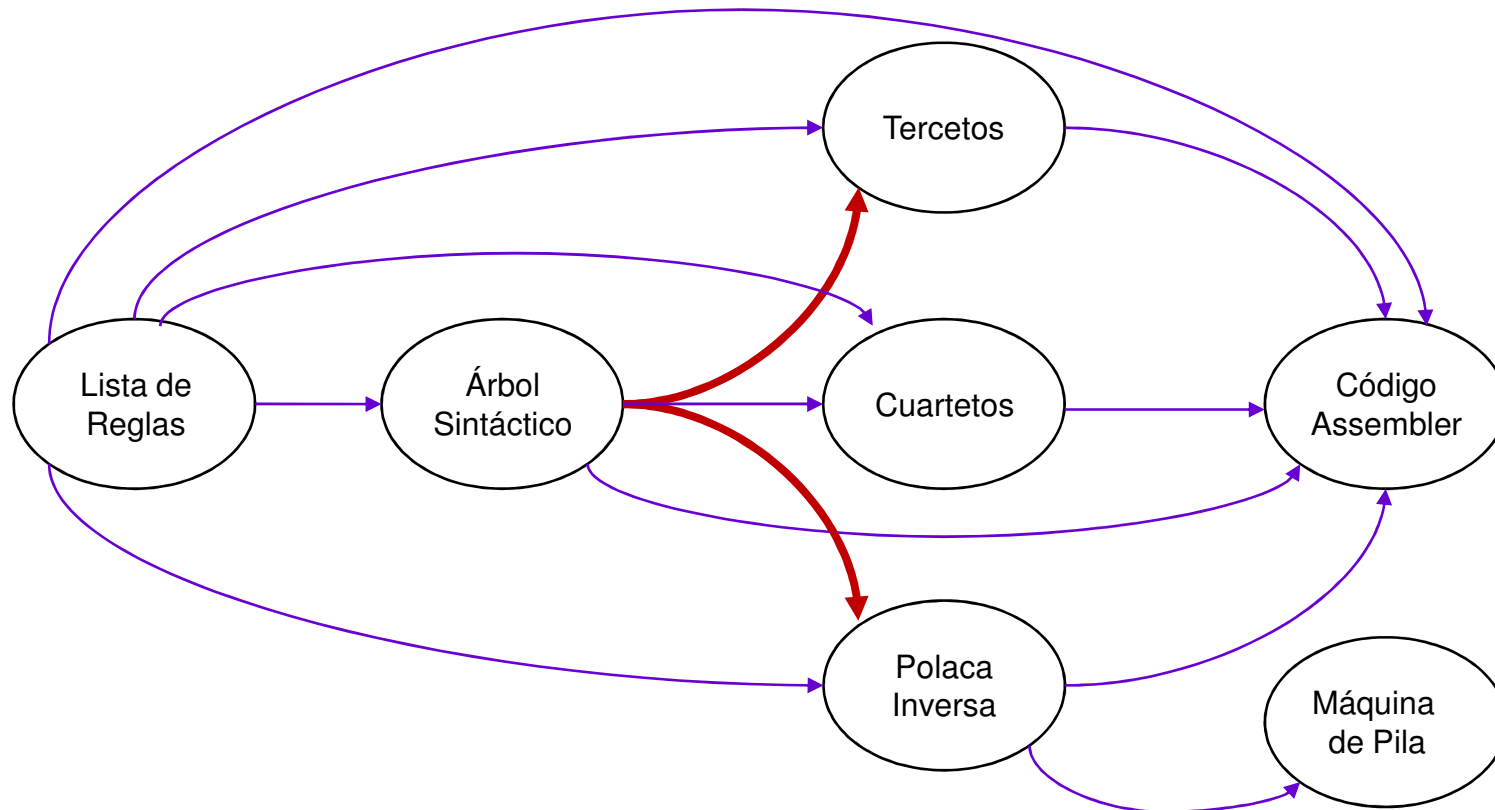
Diseño de Compiladores I

Generación de Código

Fases de la Compilación



Generación de Código



Sentencias de Control

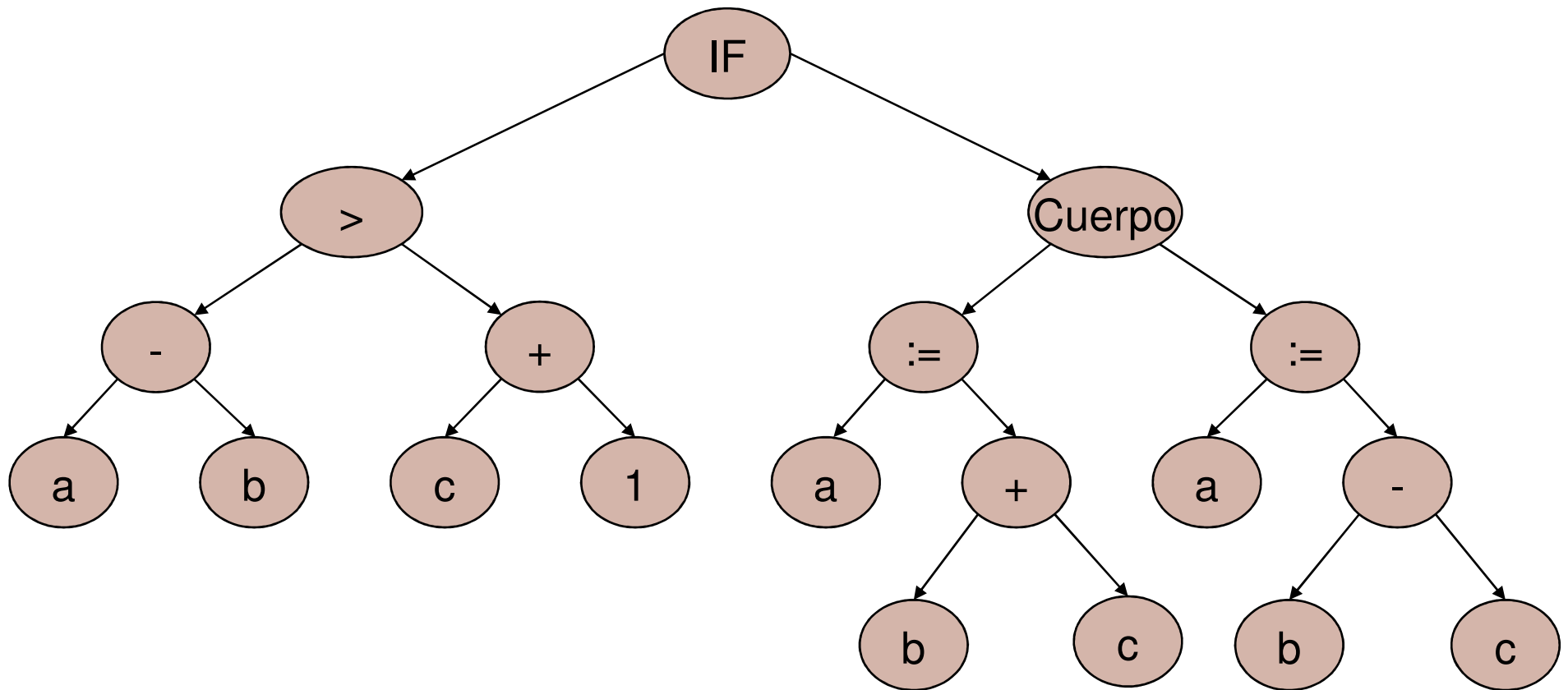
Árbol Sintáctico → Polaca Inversa

Árbol Sintáctico → Tercetos

Árbol Sintáctico \rightarrow Polaca Inversa

Ejemplo:

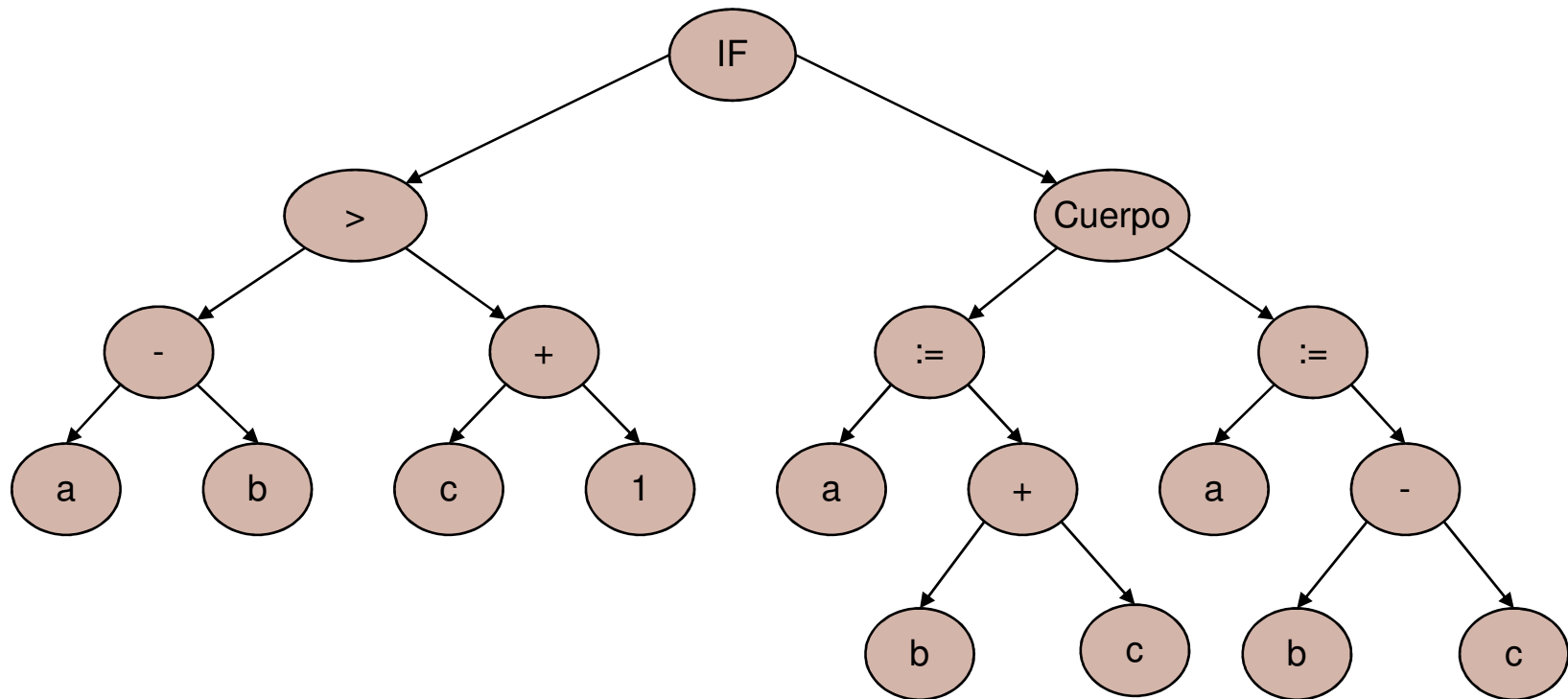
IF (a - b > c + 1) THEN a := b + c; ELSE a := b - c;



Generación de Código

Árbol Sintáctico \rightarrow Polaca Inversa

~~IF (a - b > c + 1) THEN a := b + c; ELSE a := b - c;~~



a	b	-	c		+	>	a	b	c	+	:=	a	b	c	-	:=	Cpo	IF
---	---	---	---	--	---	---	---	---	---	---	----	---	---	---	---	----	-----	----

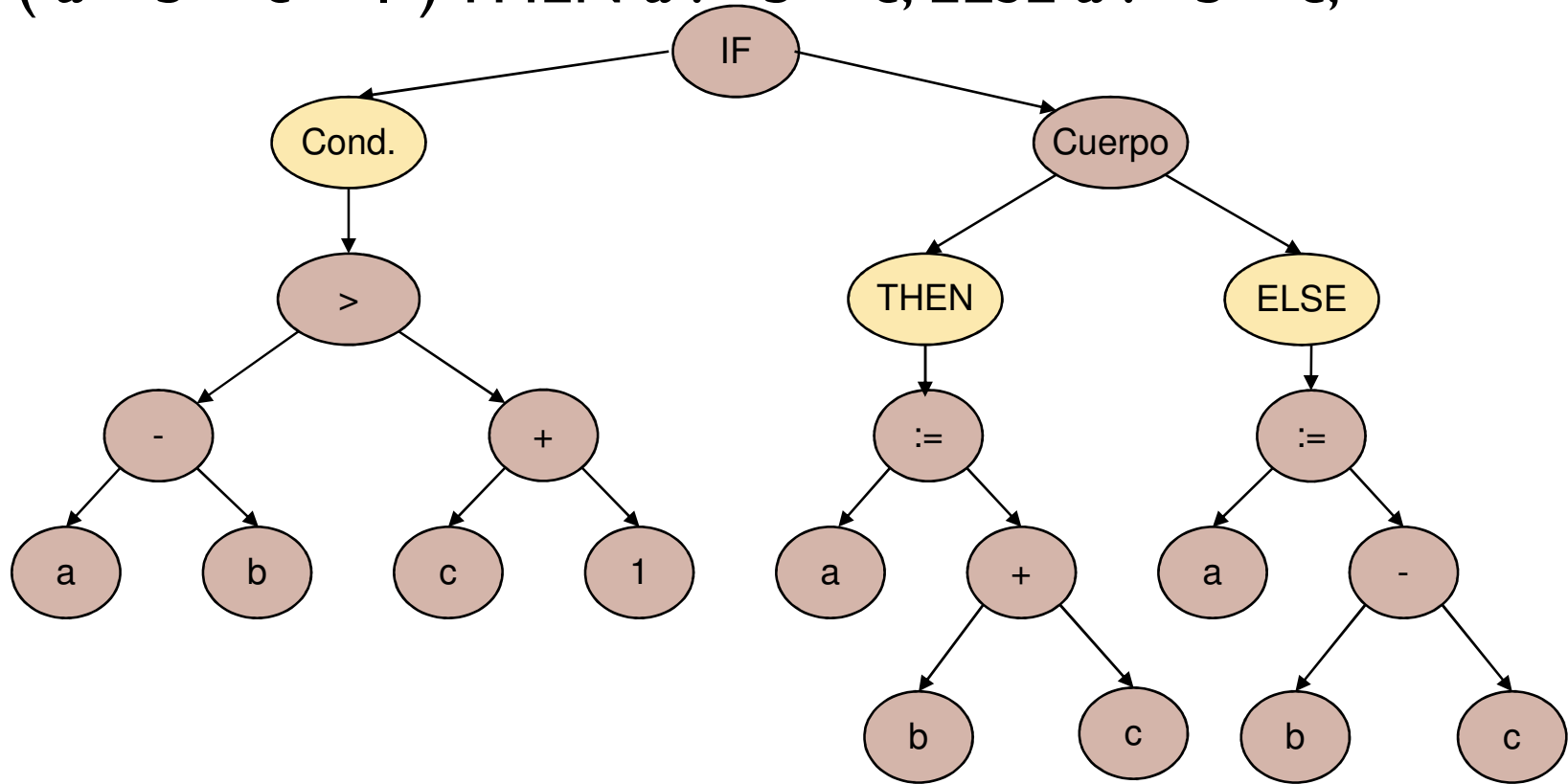
No es posible detectar inicio de THEN y ELSE para generar bifurcaciones



Generación de Código

Árbol Sintáctico → Polaca Inversa

IF (a - b > c + 1) THEN a := b + c; ELSE a := b - c;



Se agregan nodos de control para el algoritmo de generación de bifurcaciones

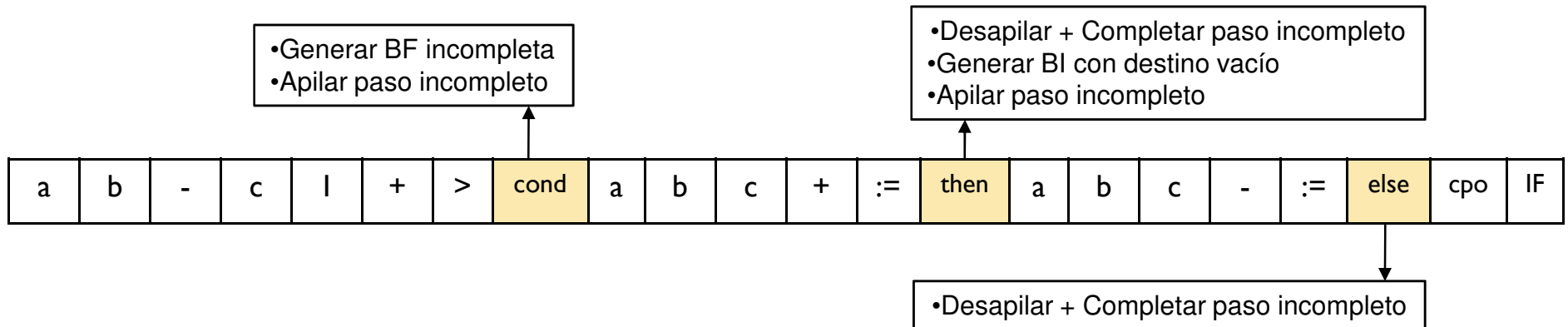
a	b	-	c		+	>	cond	a	b	c	+	:=	then	a	b	c	-	:=	else	cpo	IF
---	---	---	---	--	---	---	------	---	---	---	---	----	------	---	---	---	---	----	------	-----	----



Generación de Código

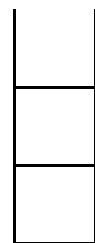
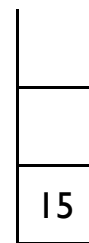
Árbol Sintáctico → Polaca Inversa

IF (a - b > c + I) THEN a := b + c; ELSE a := b - c;



Nodos de control para el algoritmo de generación de bifurcaciones

a	b	-	c	I	+	>	17	BF	a	b	c	+	:=	22	BI	a	b	c	-	:=	...
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22



Generación de Código

Árbol Sintáctico → Polaca Inversa

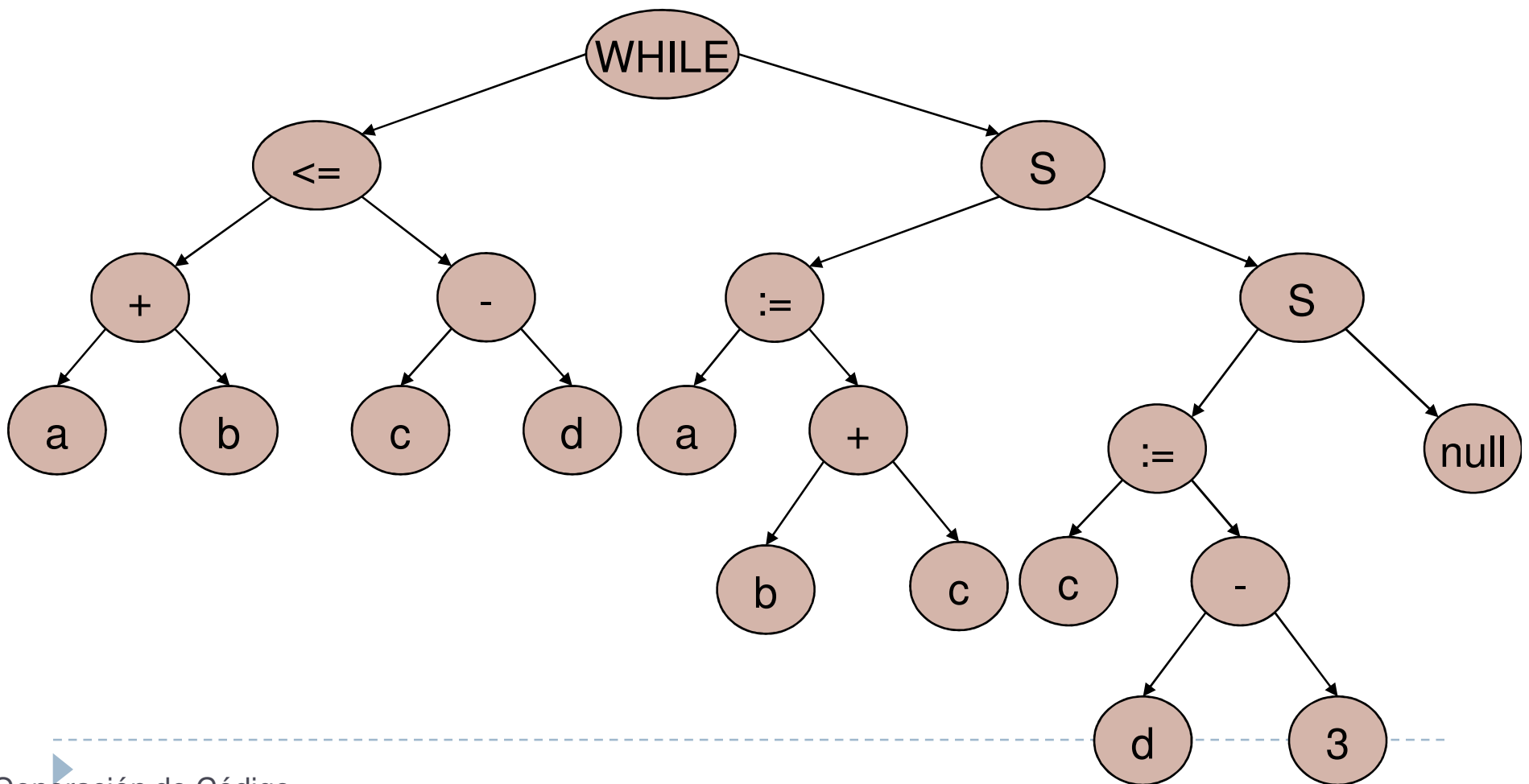
Sentencia WHILE

Generación de Código

Árbol Sintáctico → Polaca Inversa

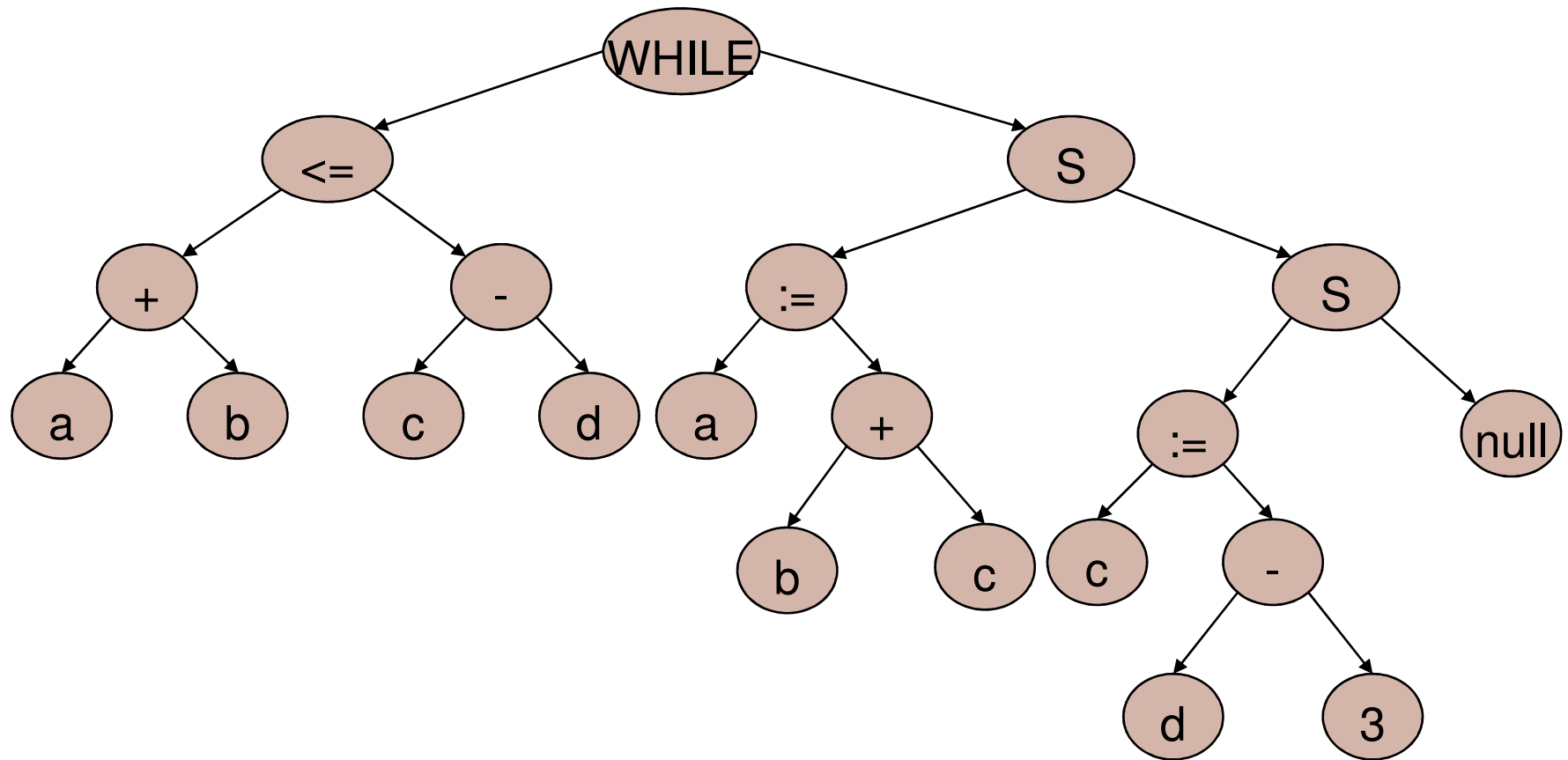
Ejemplo:

WHILE (a + b <= c - d) { a := b + c; c := d - 3; }



Árbol Sintáctico → Polaca Inversa

~~WHILE (a + b <= c - d) { a := b + c ; c := d - 3 ; }~~



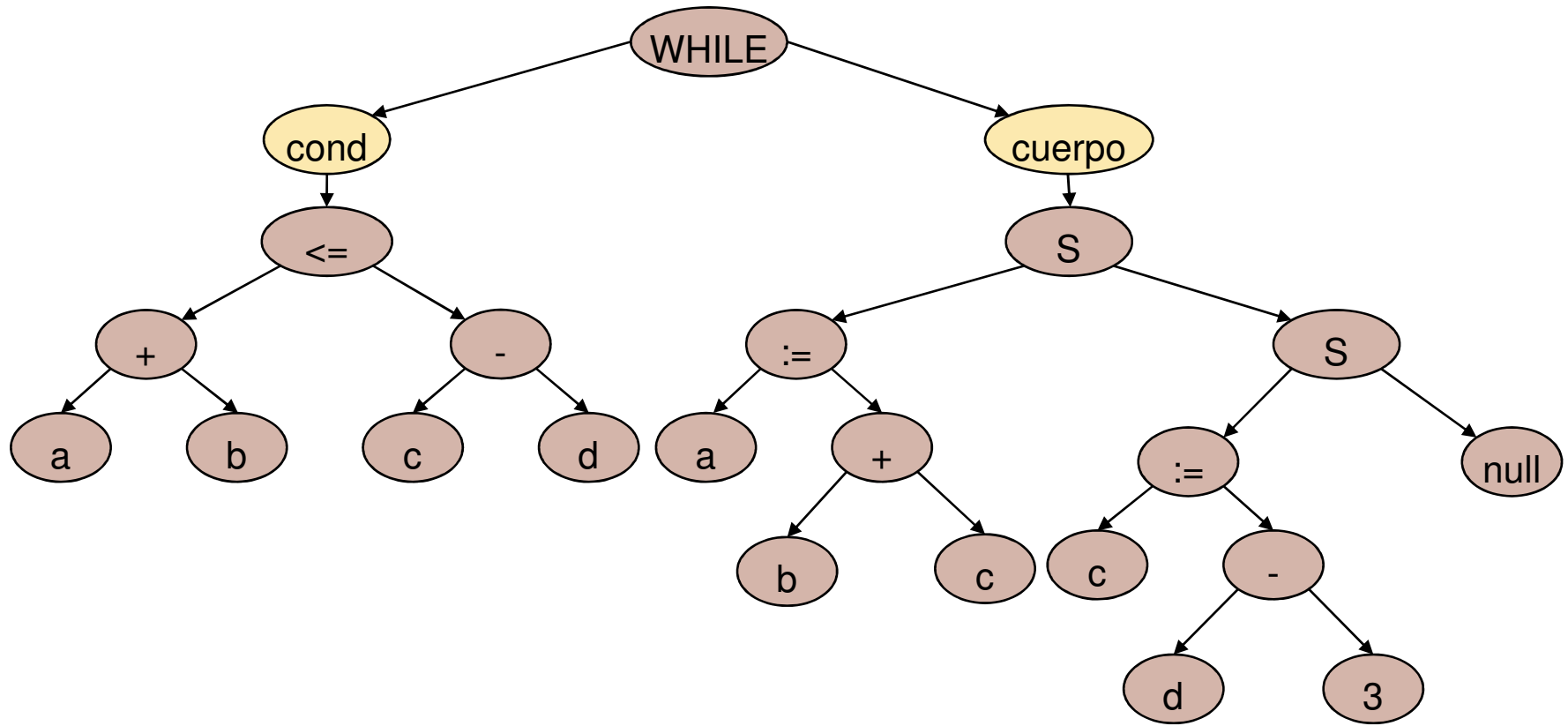
a	b	+	c	d	-	<=	a	b	c	+	:=	c	d	3	-	:=
---	---	---	---	---	---	----	---	---	---	---	----	---	---	---	---	----

~~No es posible detectar inicio y fin de bloque para generar bifurcaciones~~



Árbol Sintáctico → Polaca Inversa

WHILE (a + b <= c - d) { a := b + c; c := d - 3; }



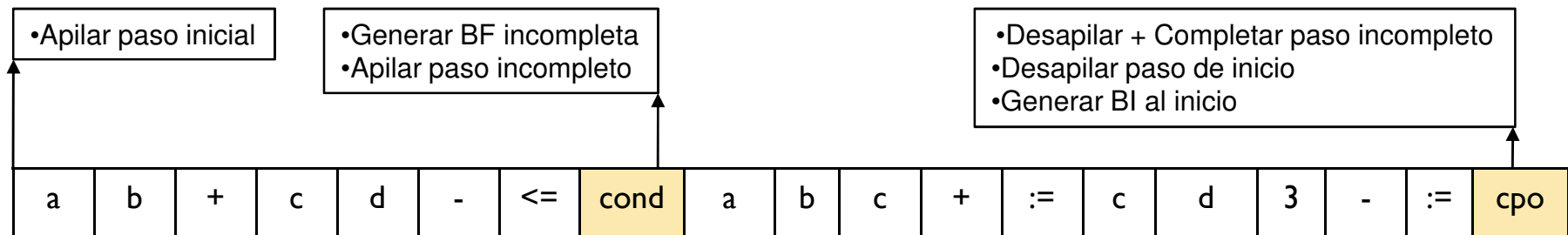
Se agregan nodos de control para el algoritmo de generación de bifurcaciones

a	b	+	c	d	-	<=	cond	a	b	c	+	:=	c	d	3	-	:=	cpo
---	---	---	---	---	---	----	------	---	---	---	---	----	---	---	---	---	----	-----



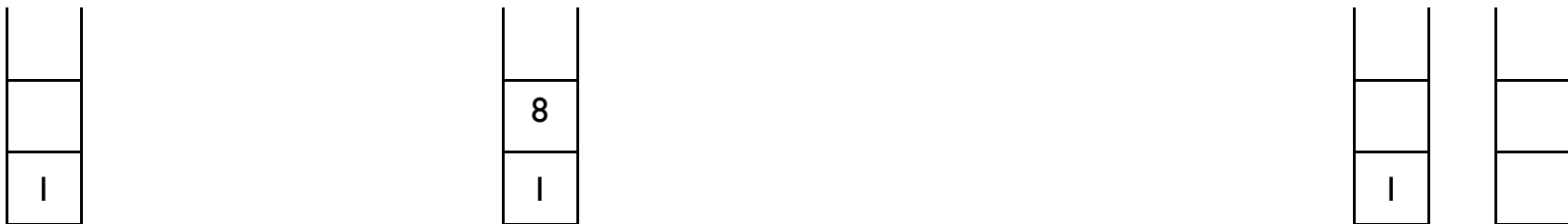
Árbol Sintáctico → Polaca Inversa

WHILE (a + b <= c - d) { a := b + c ; c := d - 3 ; }



Nodos de control para el algoritmo de generación de bifurcaciones

a	b	+	c	d	-	<=	22	BF	a	b	c	+	:=	c	d	3	-	:=	I	BI	...
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22



Árbol Sintáctico → Tercetos

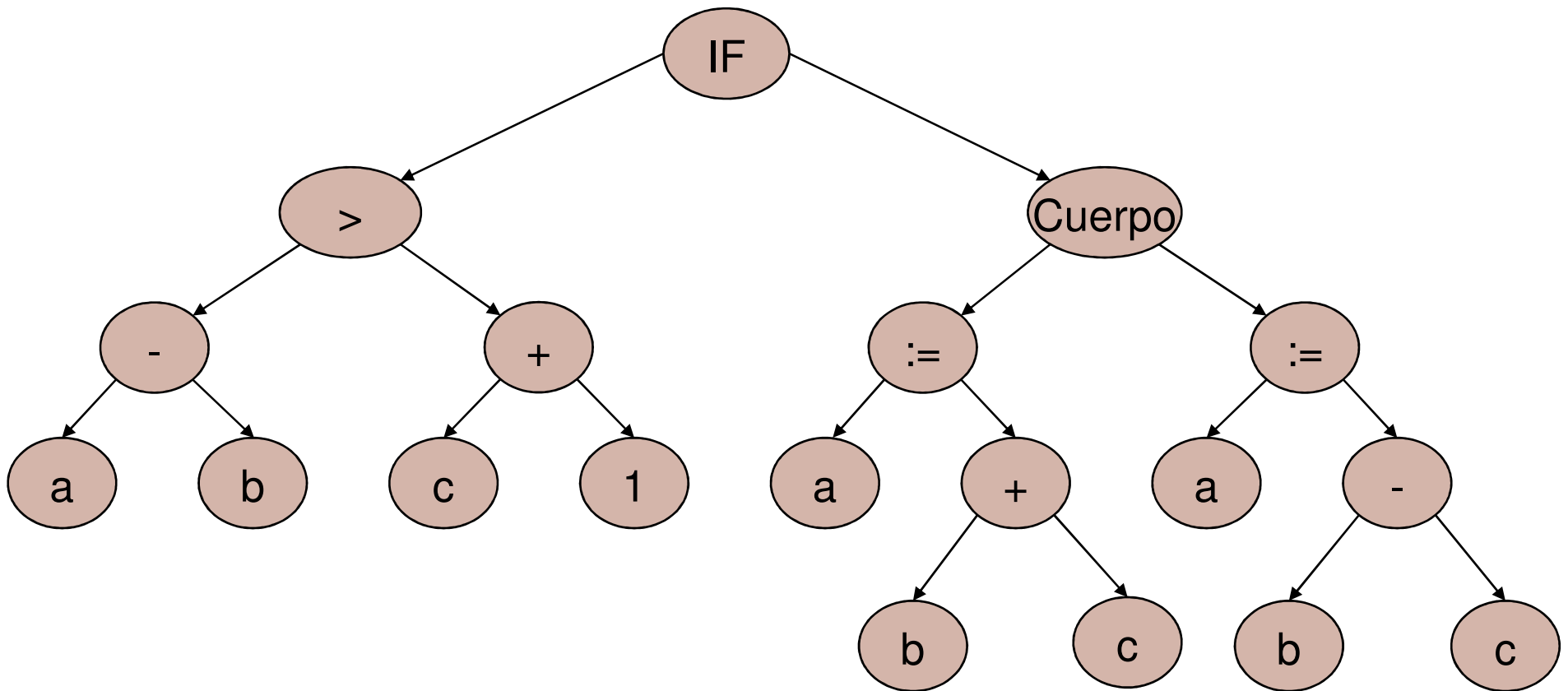
Sentencia IF

Generación de Código

Árbol Sintáctico → Tercetos

Ejemplo:

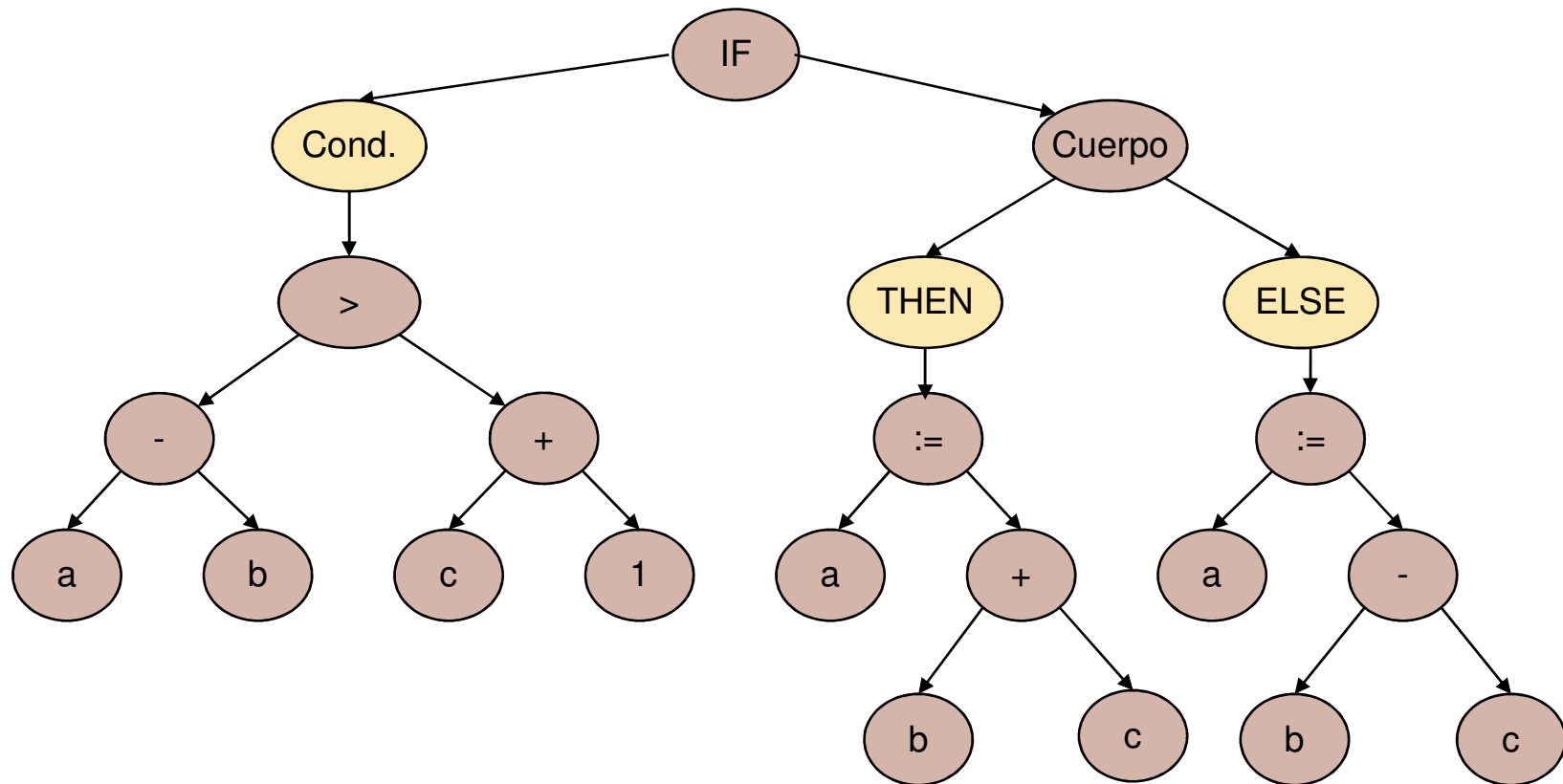
IF (a - b > c + 1) THEN a := b + c; ELSE a := b - c;



Generación de Código

Árbol Sintáctico → Tercetos

IF ($a - b > c + 1$) THEN $a := b + c$; ELSE $a := b - c$;

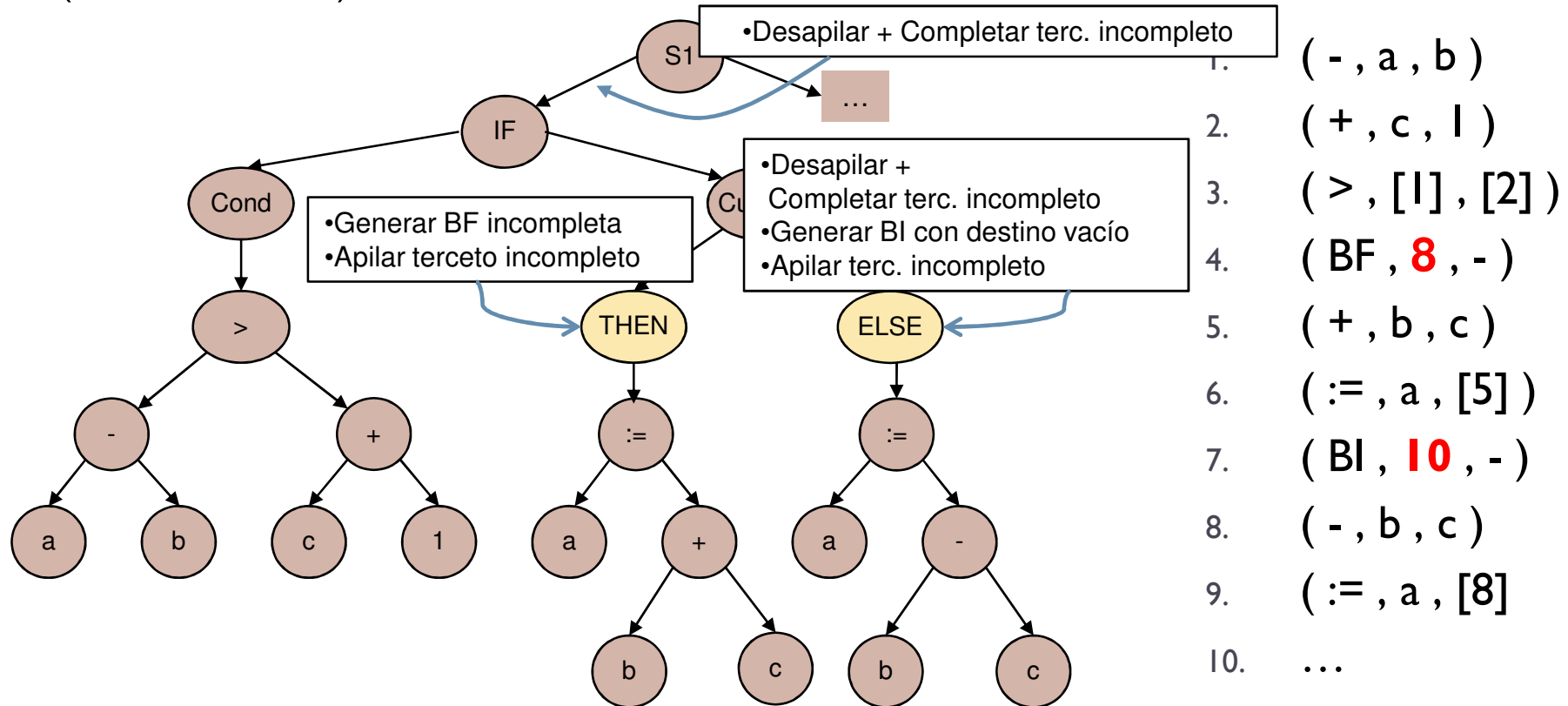


Se pueden agregar nodos de control para el algoritmo de generación de bifurcaciones

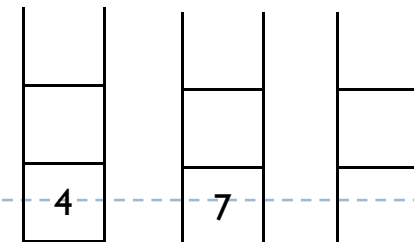


Árbol Sintáctico → Tercetos

IF (a - b > c + 1) THEN a := b + c; ELSE a := b - c;



- ▶ Se recorre el árbol en pre-orden, y se genera terceto para el subárbol de más a la izquierda con hijos hojas.
- ▶ Los nodos THEN (o hijo izq de cuerpo) y ELSE (o hijo der. de cuerpo) son los puntos de control para el algoritmo de generación de bifurcaciones.



Generación de Código

Árbol Sintáctico → Tercetos

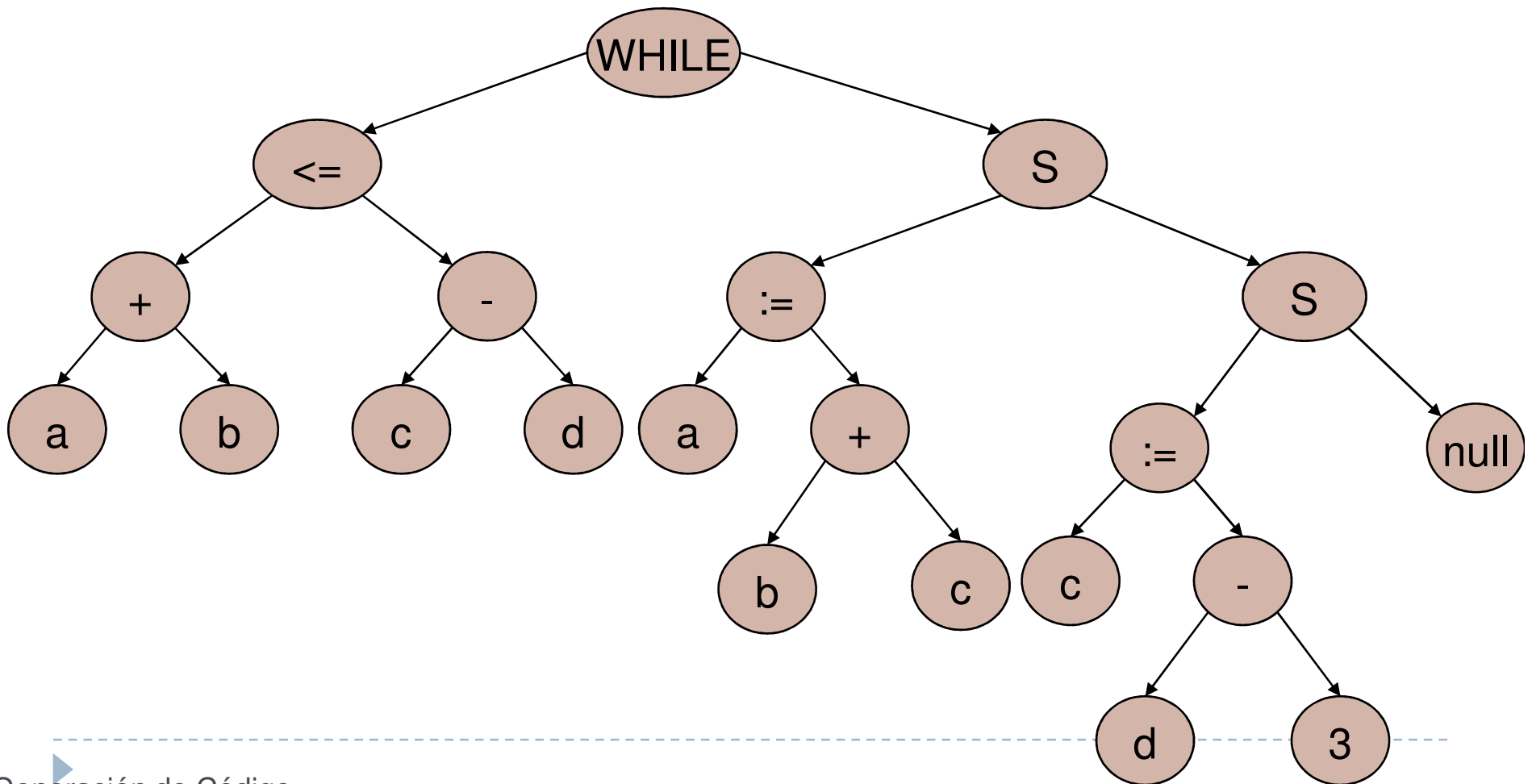
Sentencia WHILE

Generación de Código

Árbol Sintáctico → Tercetos

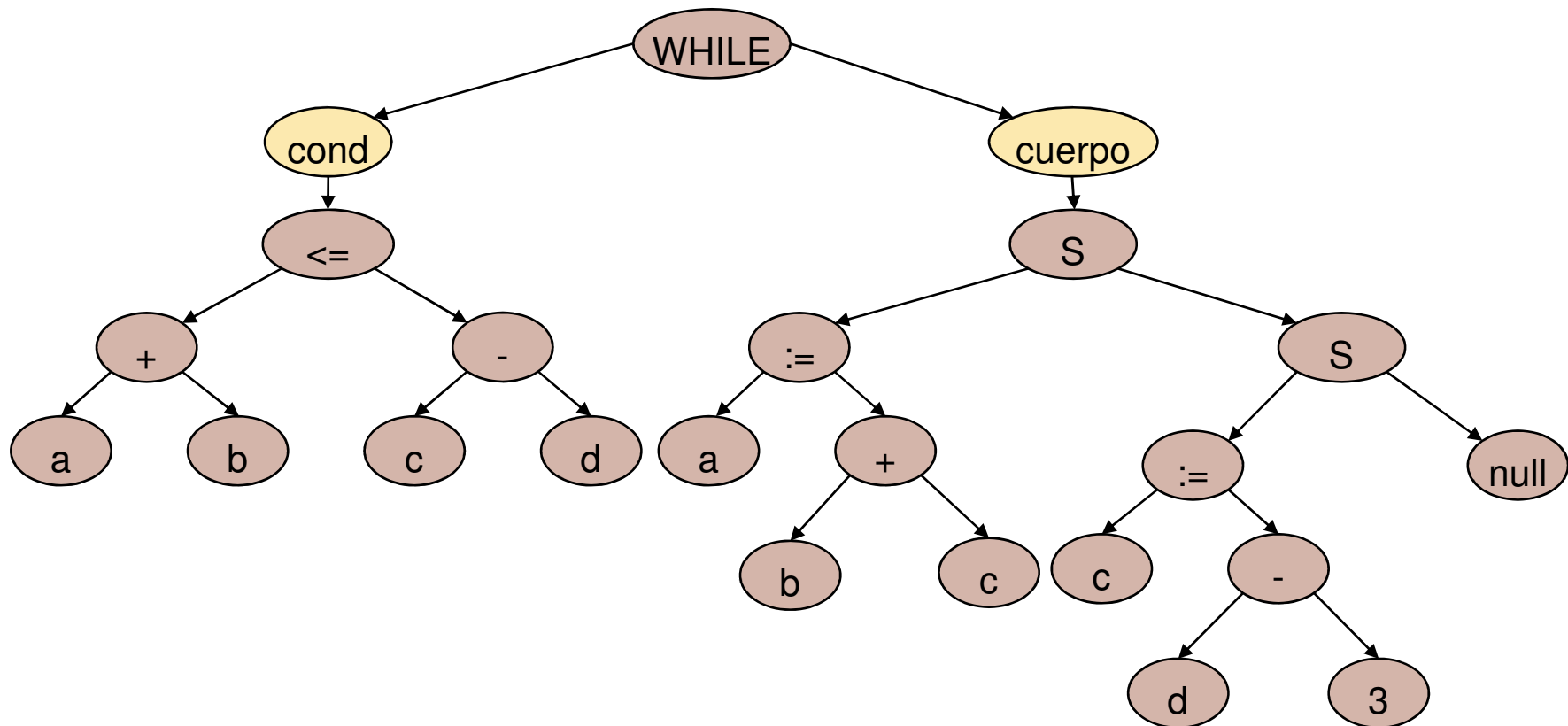
Ejemplo:

`WHILE (a + b <= c - d) { a := b + c; c := d - 3; }`



Árbol Sintáctico → Tercetos

`WHILE (a + b <= c - d) { a := b + c; c := d - 3; }`

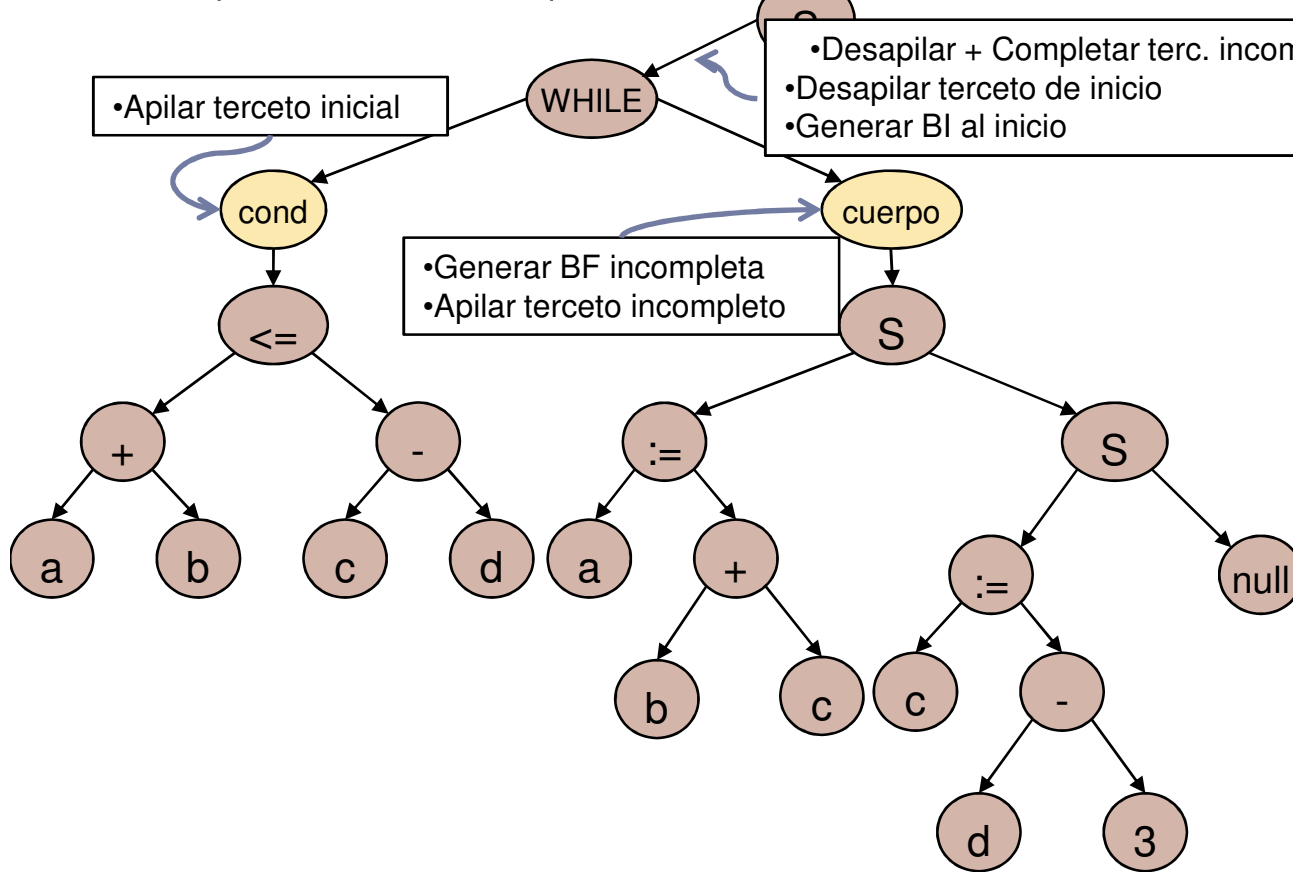


Se pueden agregar nodos de control para el algoritmo de generación de bifurcaciones



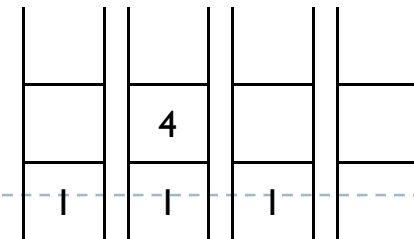
Árbol Sintáctico → Tercetos

WHILE (a + b <= c - d) { a := b + c ; c := d - 3 ; }



1. (+ , a , b)
2. (- , c , d)
3. (<= , [1] , [2])
4. (BF , **I**0 , -)
5. (+ , b , c)
6. (:= , a , [5])
7. (- , d , 3)
8. (:= , c , [7])
9. (BI , **I** , -)
10. (...)

- ▶ Se recorre el árbol en pre-orden, y se genera terceto para el subárbol de más a la izquierda con hijos hojas.
- ▶ Los nodos **cond** (o hijo izq de WHILE) y **cuerpo** (o hijo der. de WHILE) son los puntos de control para el algoritmo de generación de bifurcaciones.



Generación de Código

Ejercicios

- ▶ Para un lenguaje que permite la sentencia DO UNTIL

- ▶ Considerando el camino

Lista de Reglas → Árbol Sintáctico → Polaca Inversa

1. ¿Cuáles serán las representaciones intermedias para la siguiente sentencia?

```
DO
BEGIN
    a := a * 2;
    x := a;
END
UNTIL ( x > 10)
```

2. ¿Cuáles serán los puntos de control en el recorrido del árbol, para generar las bifurcaciones en la Polaca Inversa?



Ejercicios

- ▶ Para un lenguaje que permite la sentencia FOR
 - ▶ Considerando el camino Lista de Reglas → Tercetos
1. ¿Cuáles serán las representaciones intermedias para la siguiente sentencia?

```
FOR (i := 0; i < 10; i++)  
BEGIN  
    a := a * 2;  
    x := a;  
END
```

2. ¿Cuáles serán los puntos de control en el recorrido del árbol, para generar las bifurcaciones en los tercetos?



¿PREGUNTAS?