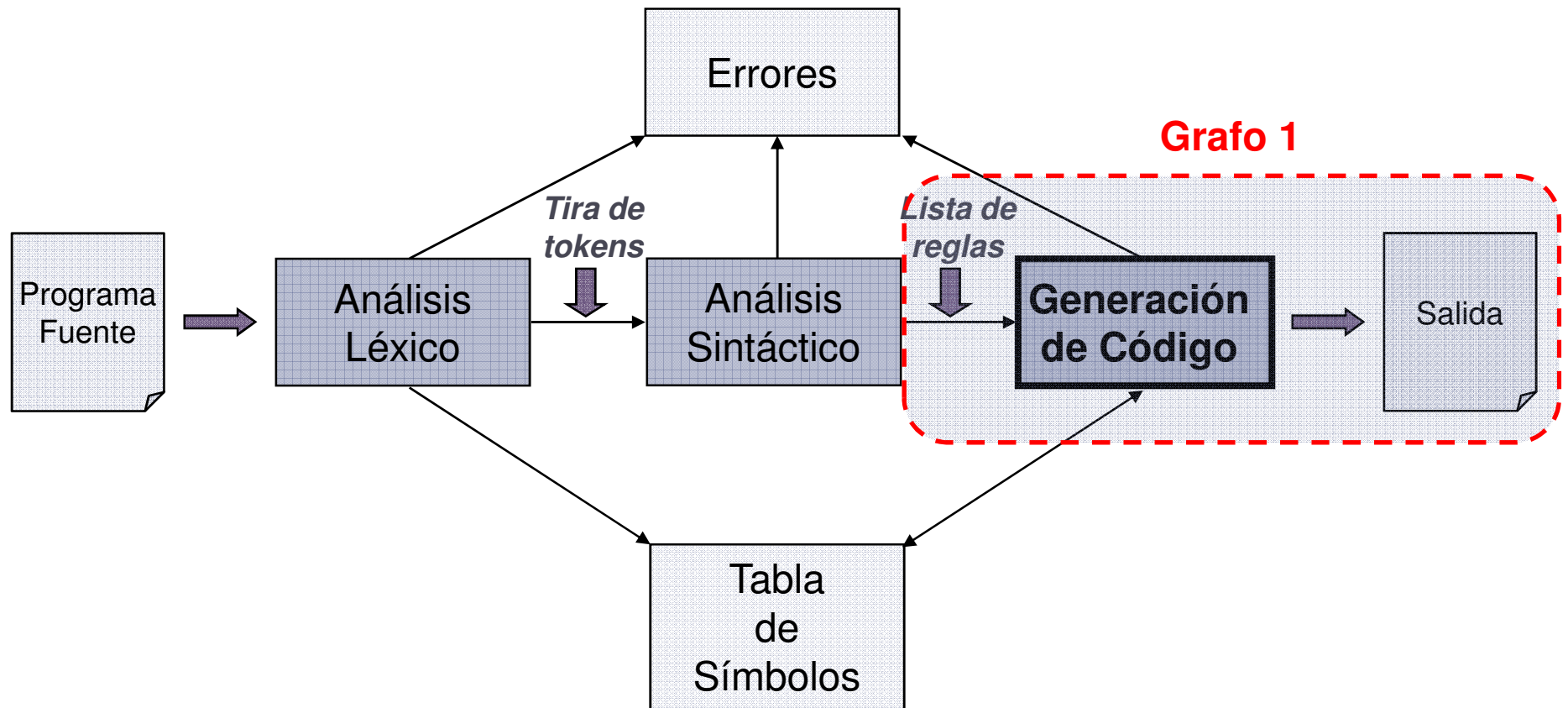


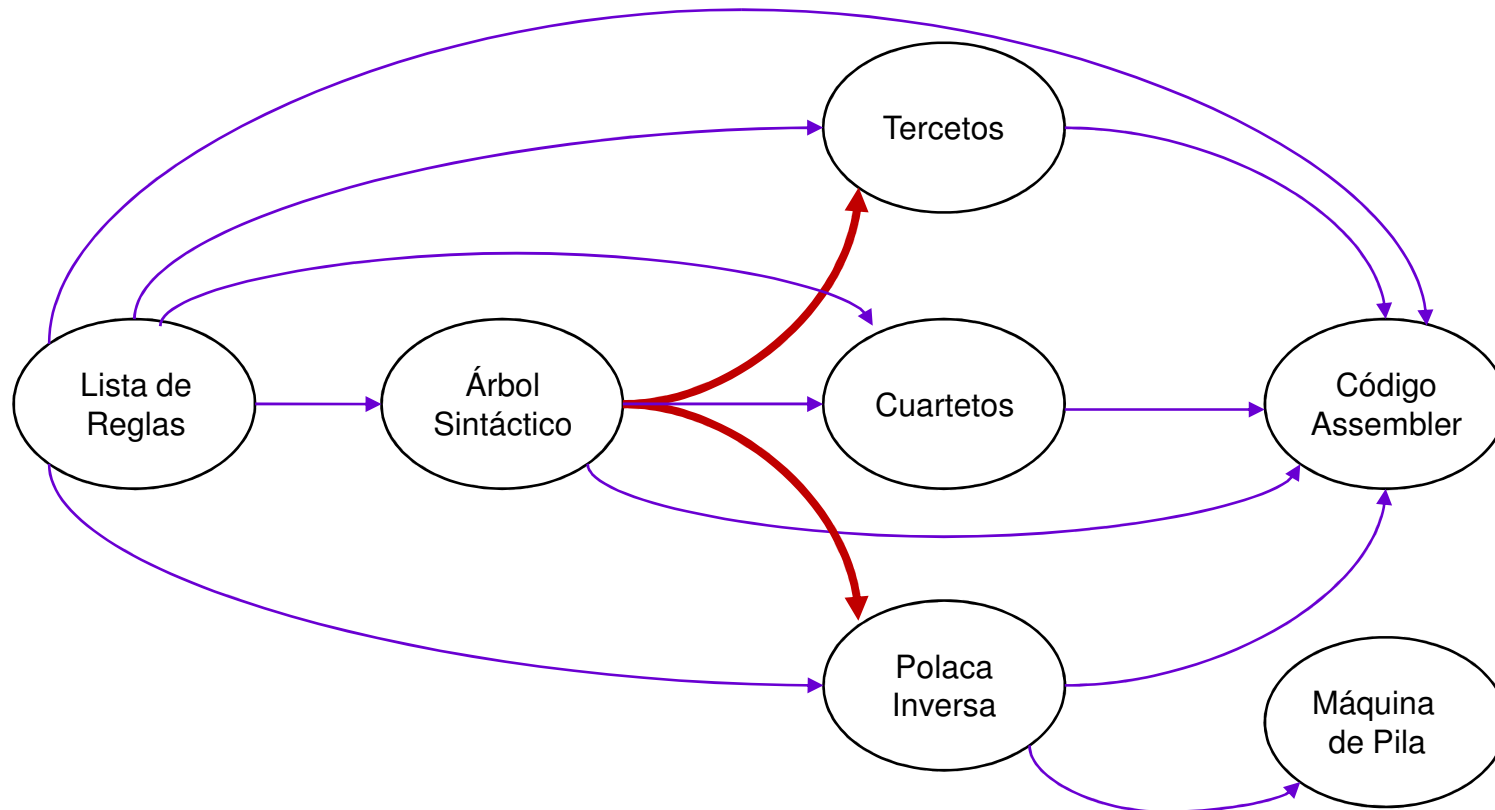
Diseño de Compiladores I

Generación de Código

Fases de la Compilación



Generación de Código



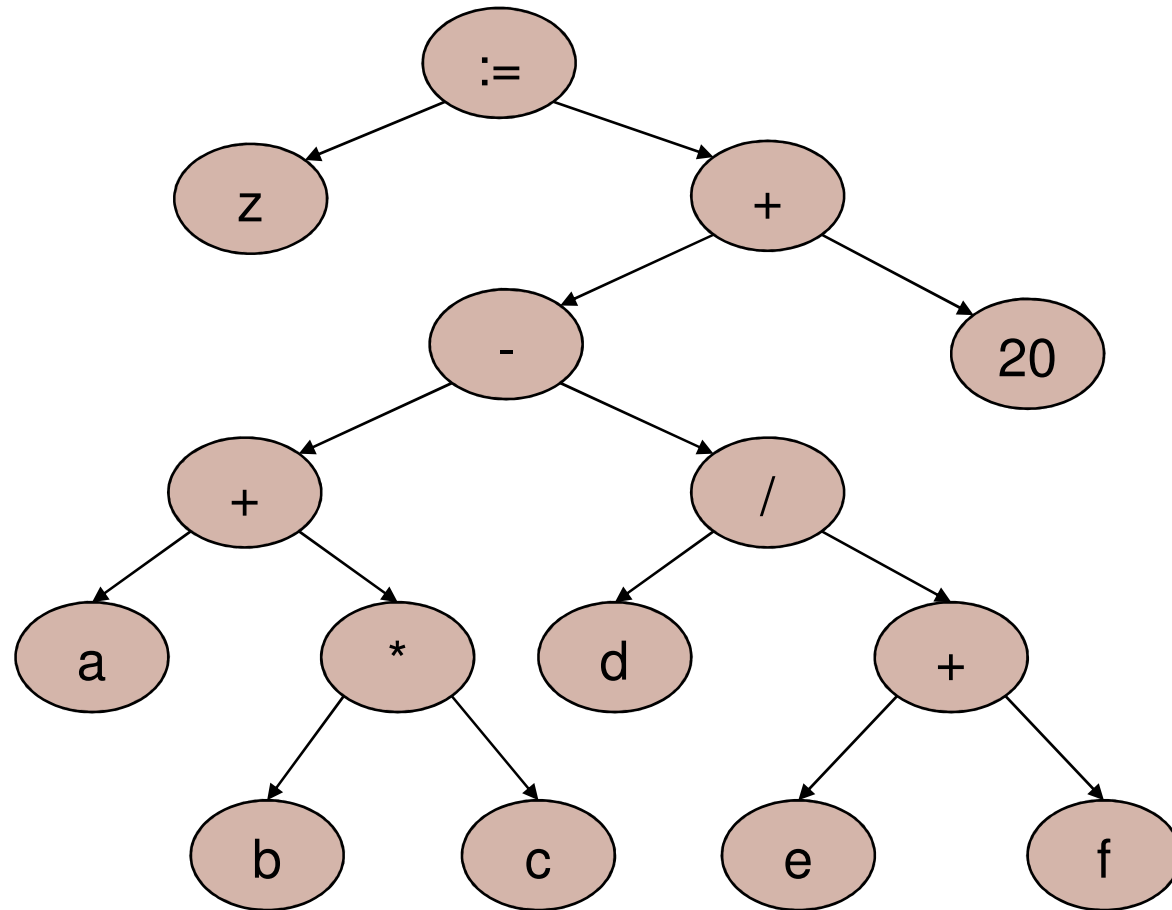
Recorridos del Árbol Sintáctico

- ▶ Inorden → Código original
- ▶ Postorden → Polaca Inversa
- ▶ Preorden → Tercetos



Recorridos del Árbol Sintáctico

Ejemplo: $z := a + b * c - d / (e + f) + 20.$



Generación de Código

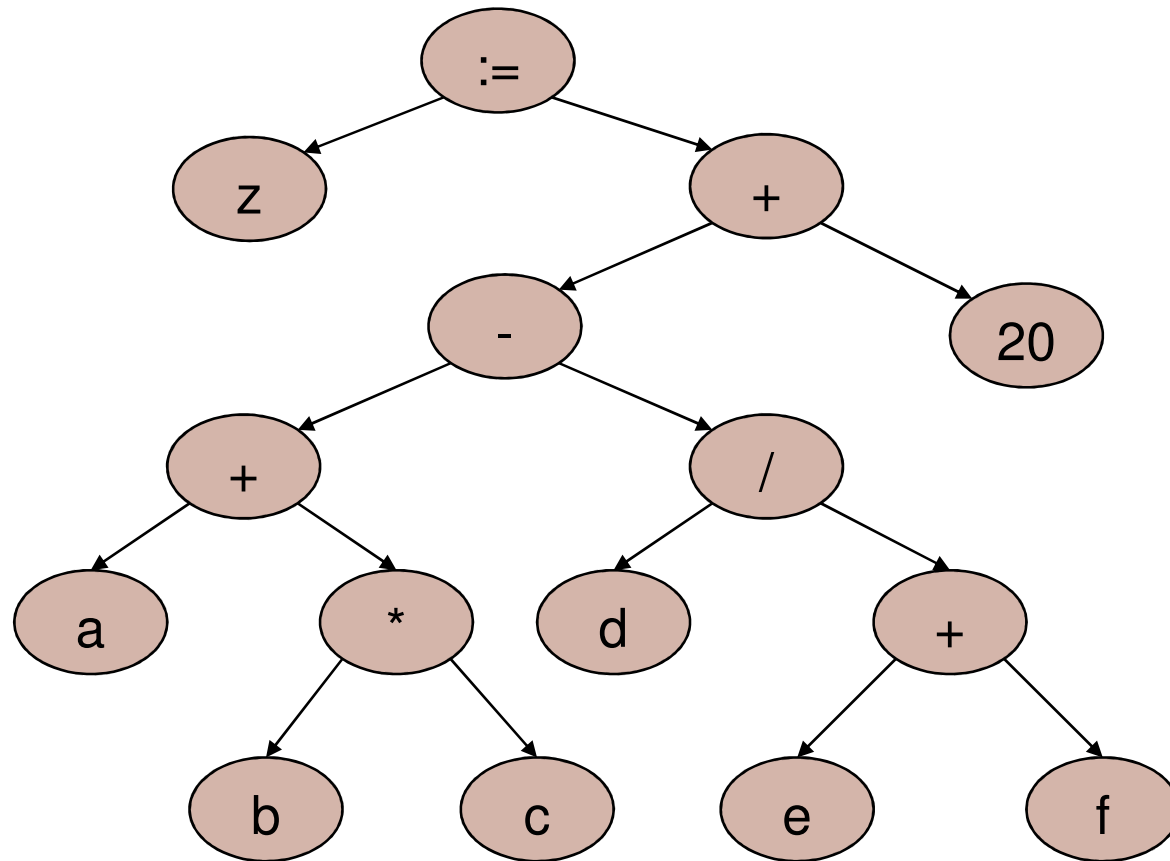
ASIGNACIONES / EXPRESIONES

Árbol Sintáctico → Polaca Inversa

Árbol Sintáctico → Tercetos

Árbol Sintáctico → Polaca Inversa

Ejemplo: $z := a + b * c - d / (e + f) + 20.$



z	a	b	c	*	+	d	e	f	+	/	-	20	+	:=
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	----------	-----------

Árbol Sintáctico → Polaca Inversa

Ejemplo: $z := a + b * c - d / (e + f) + 20$

Árbol Sintáctico → Polaca Inversa

z	a	b	c	*	+	d	e	f	+	/	-	20	+	:=
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	----------	-----------

Lista de Reglas → Polaca Inversa

a	b	c	*	+	d	e	f	+	/	-	20	+	z	:=
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	----------	----------	-----------



Árbol Sintáctico → Polaca Inversa (Tipos)

(a) (a)

Árbol Sintáctico sin tipos ni conversiones → Polaca sin tipos ni conversiones

(El chequeo de tipos y la incorporación de conversiones se posterga a la traducción Polaca Inversa → Assembler)

(b) (b)

Árbol Sintáctico con tipos → Polaca con tipos

(La incorporación de conversiones se posterga a la traducción Polaca Inversa → Assembler)

(c) (c)

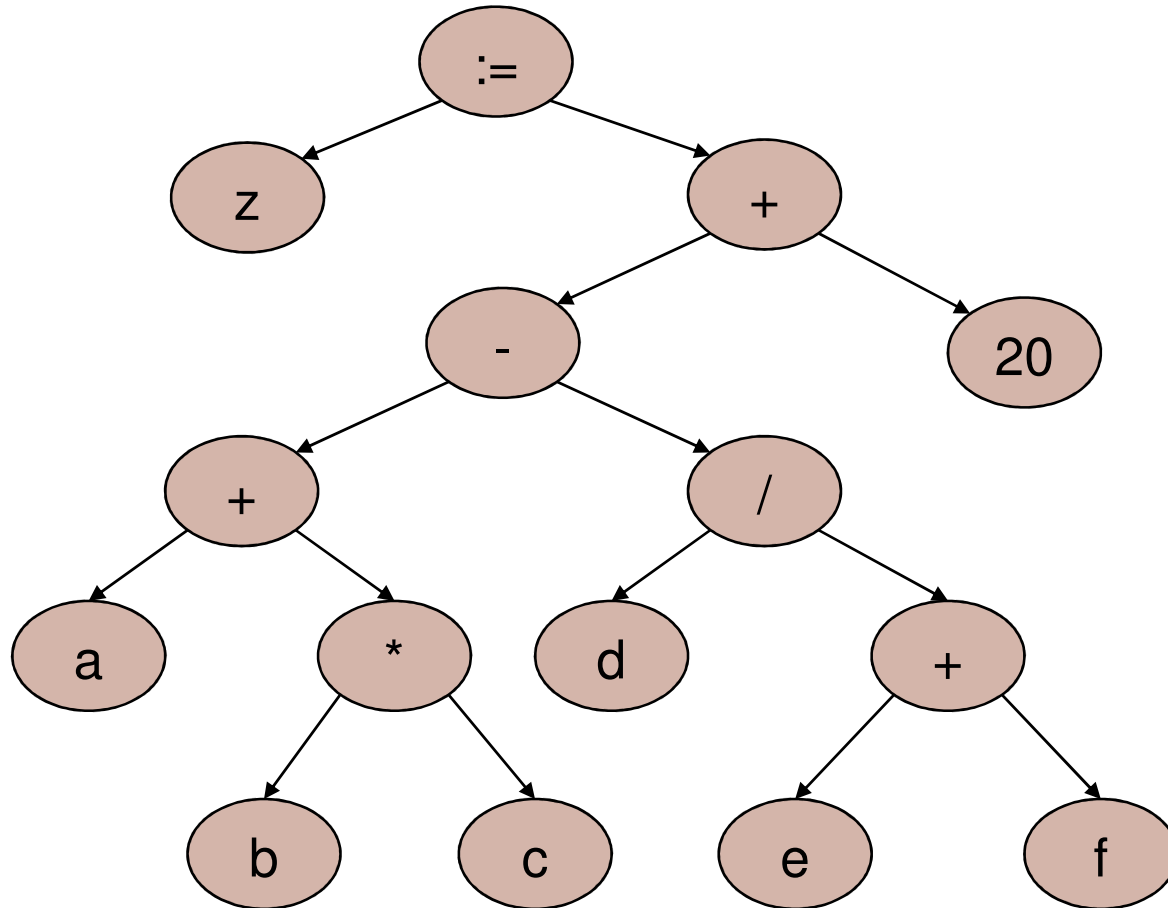
Árbol Sintáctico con tipos y conversiones → Polaca con tipos y conversiones



Árbol Sintáctico → TERCETOS

Árbol Sintáctico \rightarrow Tercetos

Ejemplo: $z := a + b * c - d / (e + f) + 20.$

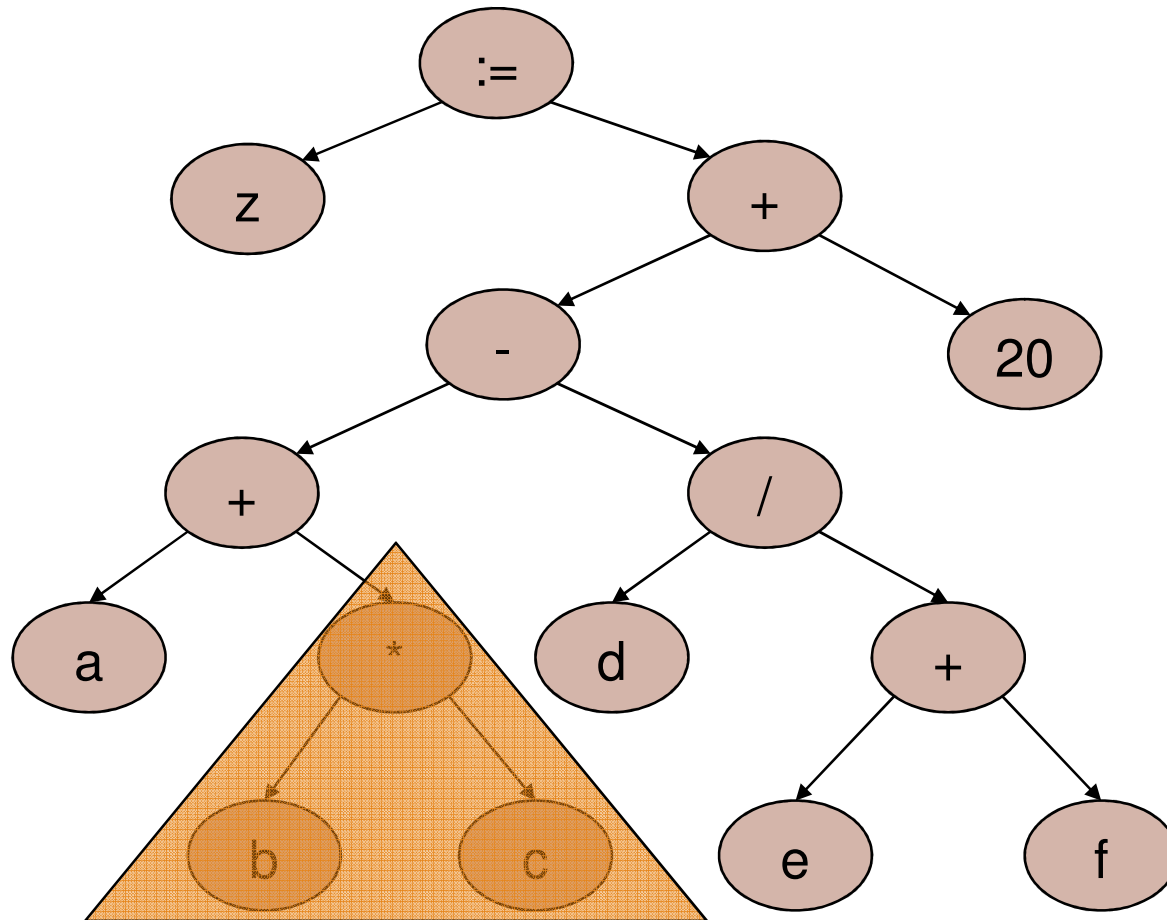


Árbol Sintáctico → Tercetos

- ▶ Se recorre el árbol, buscando el subárbol de más a la izquierda con hijos hojas.
(Si se trata de un operador unario, será un subárbol con un solo hijo hoja)
- ▶ Se genera el terceto para el subárbol.
- ▶ Se reemplaza el subárbol por el número de terceto generado.



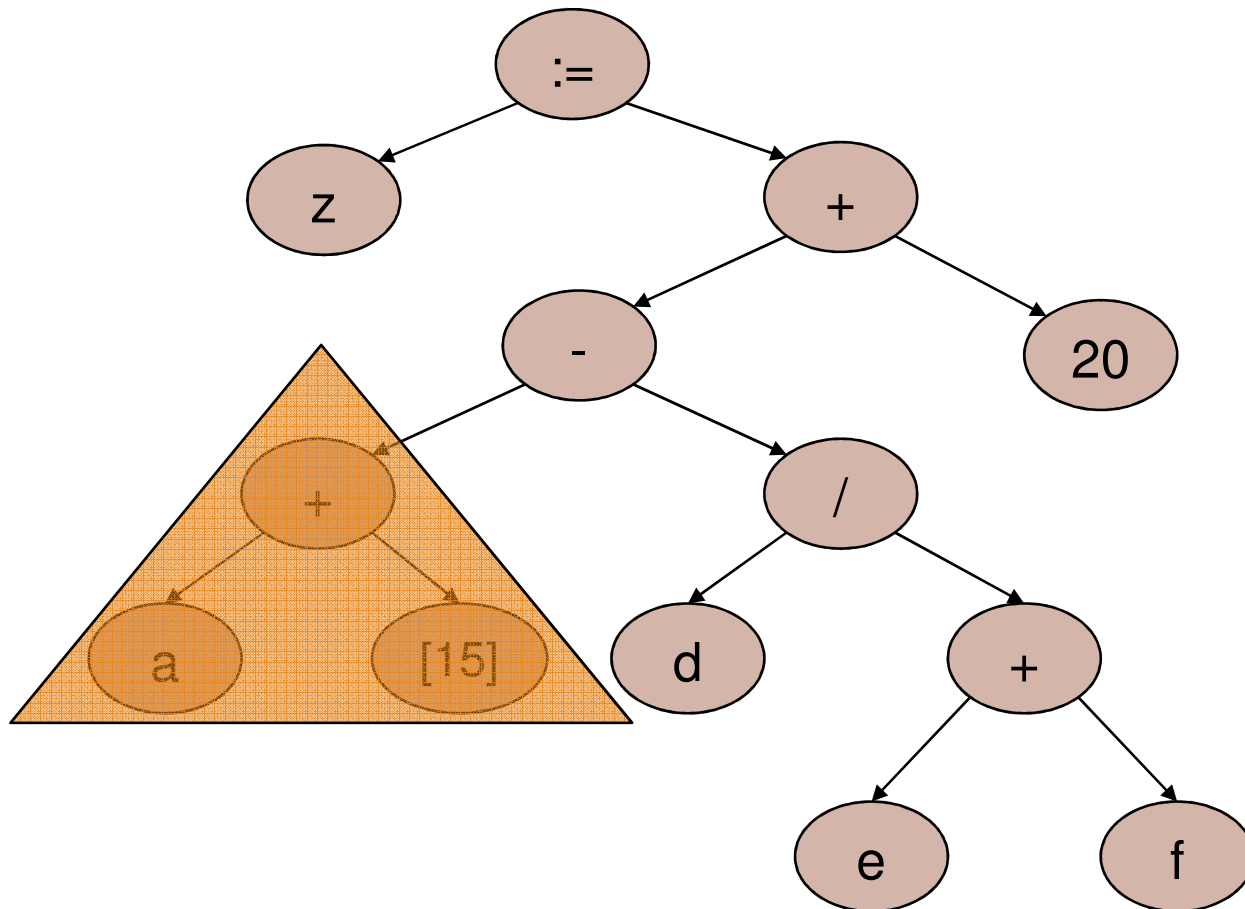
Árbol Sintáctico → Tercetos



14. ...
15. (*, b, c)



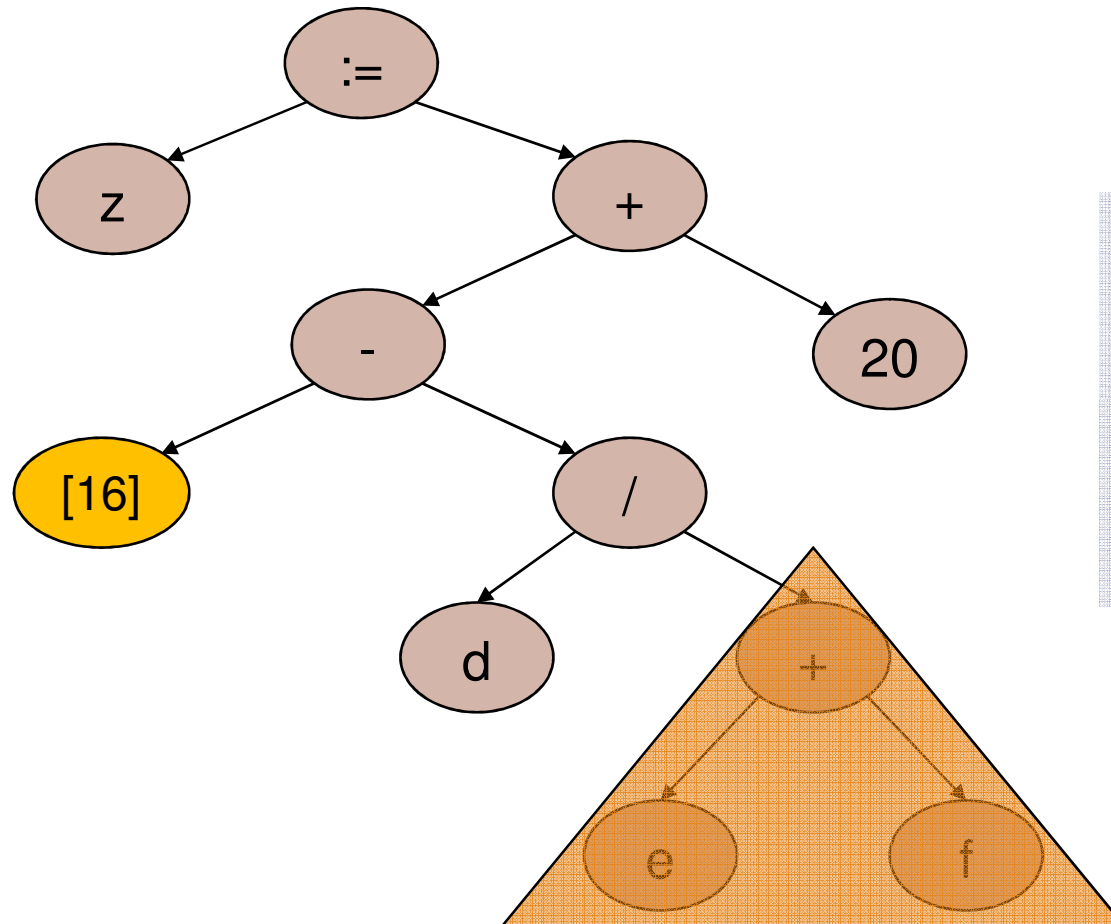
Árbol Sintáctico → Tercetos



- 14. ...
- 15. (*, b , c)
- 16. (+ , a , [15])



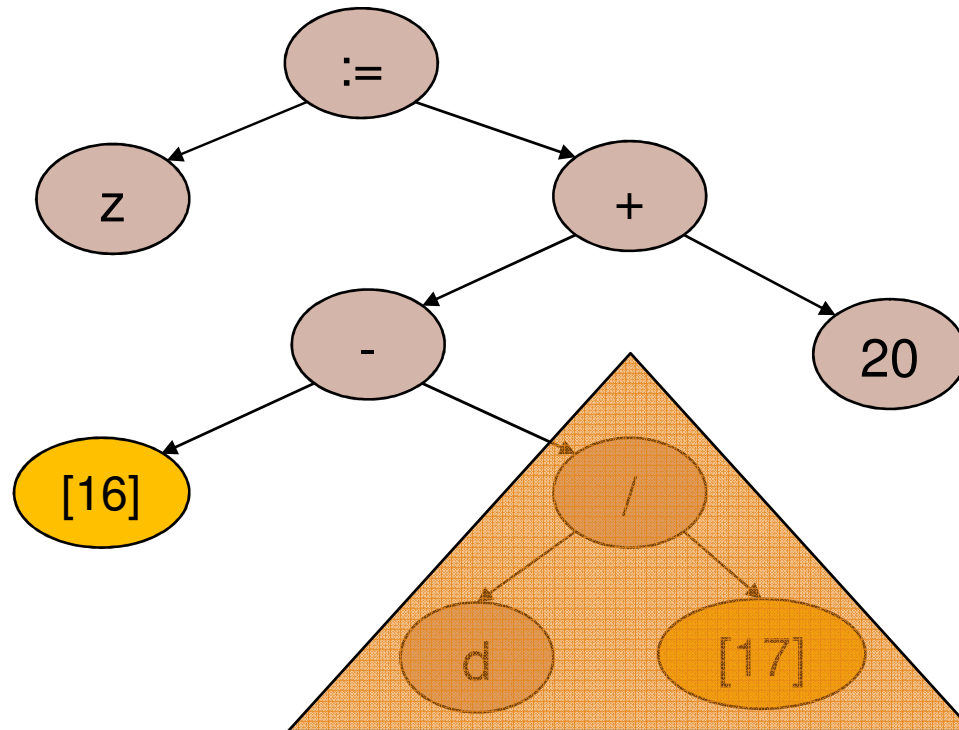
Árbol Sintáctico → Tercetos



- 14. ...
- 15. (*, b, c)
- 16. (+, a, [15])
- 17. (+, e, f)



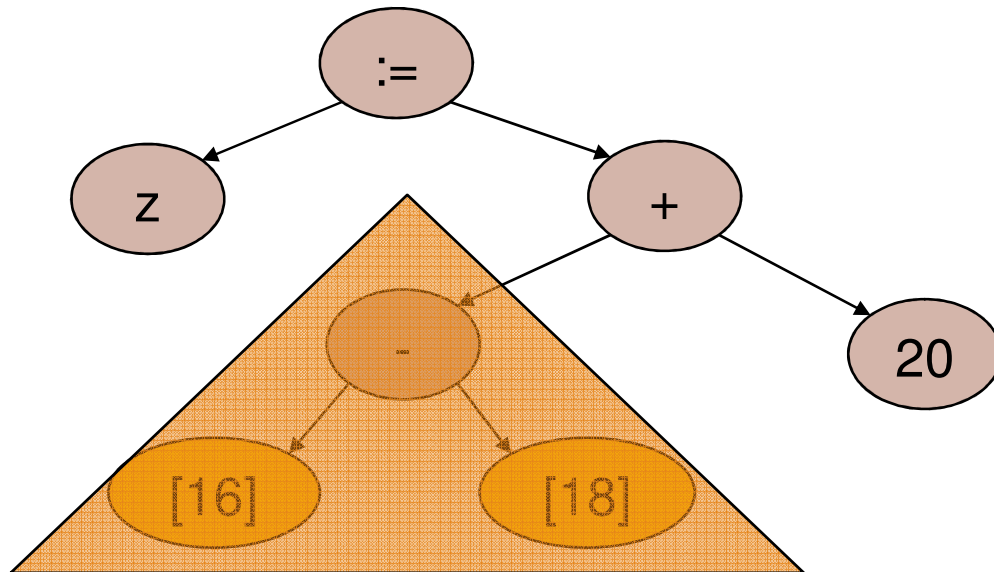
Árbol Sintáctico → Tercetos



14. ...
15. (*, b, c)
16. (+, a, [15])
17. (+, e, f)
18. (/, d, [17])



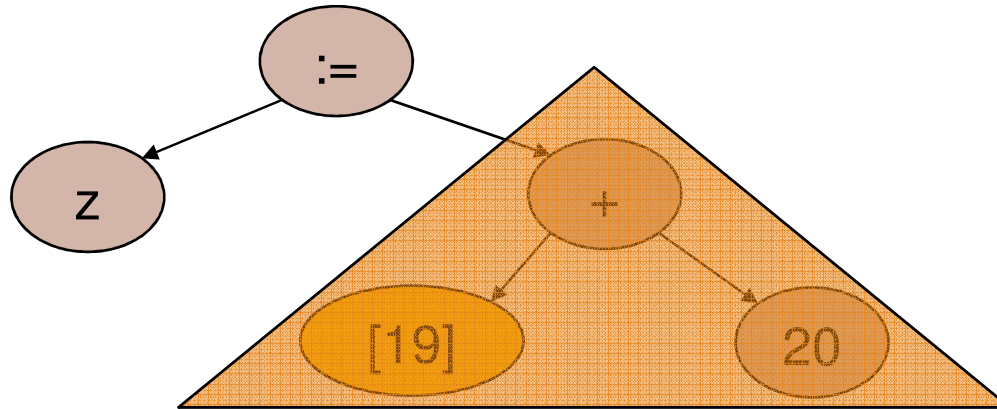
Árbol Sintáctico → Tercetos



14. ...
15. (*, b , c)
16. (+ , a , [15])
17. (+ , e , f)
18. (/ , d , [17])
19. (- , [16] , [18])

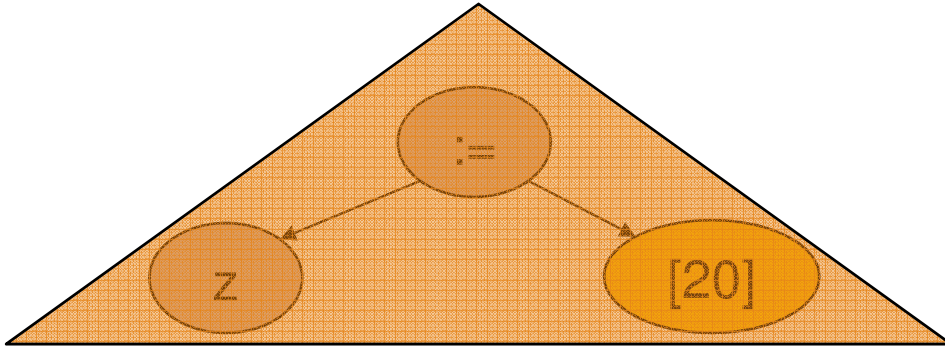


Árbol Sintáctico → Tercetos



```
14. ...
15. (*, b, c)
16. (+, a, [15])
17. (+, e, f)
18. (/ , d, [17])
19. (-, [16], [18])
20. (+, [19], 20)
```

Árbol Sintáctico → Tercetos



```
14. ...
15. ( *, b , c )
16. ( + , a , [15] )
17. ( + , e , f )
18. ( / , d , [17] )
19. ( - , [16] , [18] )
20. ( + , [19] , 20 )
21. ( := , z , [20] )
```

Árbol Sintáctico → Tercetos (Tipos)

(a) (a)

Árbol Sintáctico sin tipos ni conversiones → Tercetos sin tipos ni conversiones

El chequeo de tipos y la incorporación de conversiones se puede postergar hasta la traducción Tercetos
→ Assembler

(a) (b)

Árbol Sintáctico sin tipos ni conversiones → Tercetos con tipos

Se incorporan los tipos

La incorporación de conversiones se posterga a la traducción Tercetos → Assembler

(a) (c)

Árbol Sintáctico sin tipos ni conversiones → Tercetos con tipos y conversiones

Se incorporan los tipos

Se incorporan las conversiones

(b) (b)

Árbol Sintáctico con tipos → Tercetos con tipos

La incorporación de conversiones se puede postergar a la traducción Tercetos → Assembler

(b) (c)

Árbol Sintáctico con tipos → Tercetos con tipos y conversiones

Se incorporan las conversiones

(c) (c)

Árbol Sintáctico con tipos y conversiones → Tercetos con tipos y conversiones
