

6. Generación de Código Intermedio

➤ Introducción: necesidad de un Código Intermedio

6.1. GCI para expresiones e instrucciones

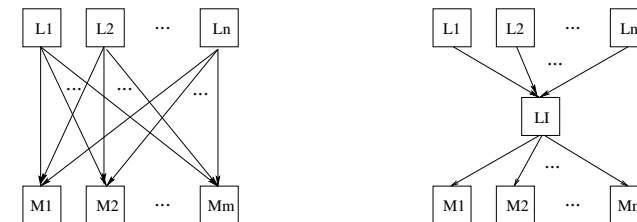
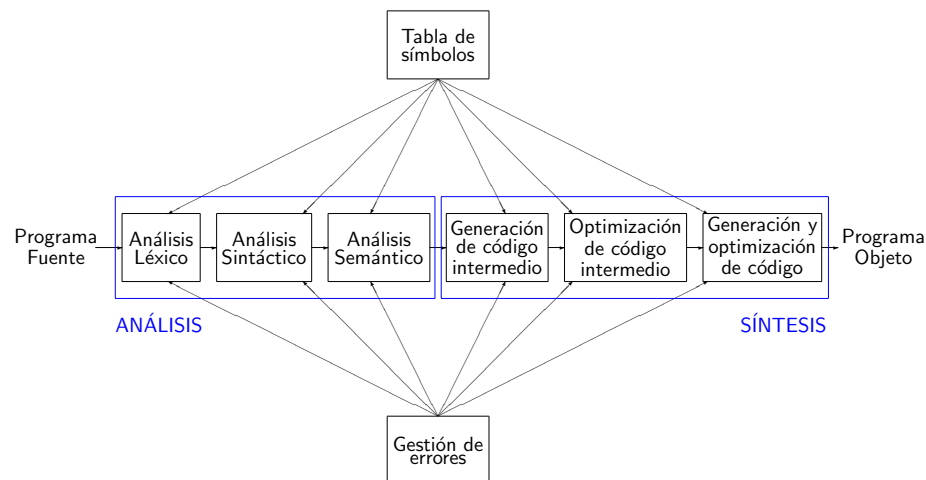
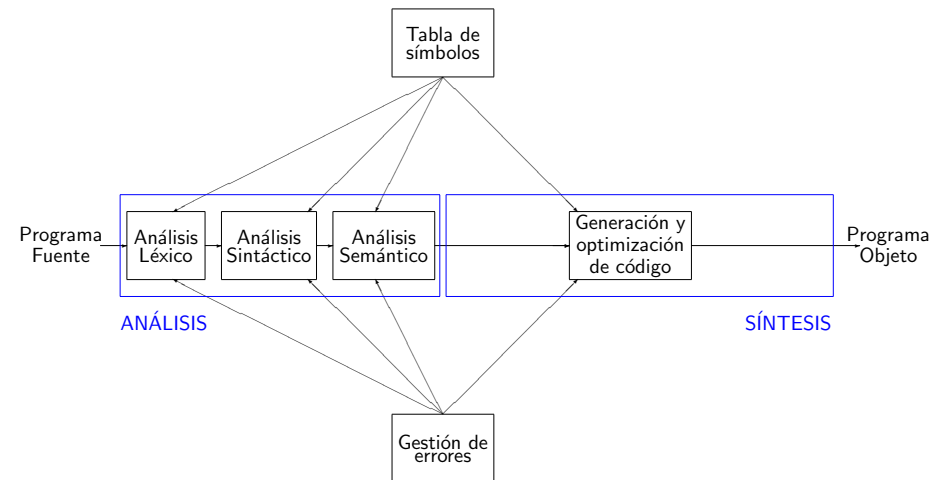
- Objetos simples
- Objetos estructurados: *registro*
- Objetos estructurados: *array*
- Expresiones lógicas

6.2. GCI para instrucciones que rompen el flujo de control

- Listas de referencias no satisfechas
- Instrucciones que rompen el flujo de control

6.3. GCI para procedimientos y funciones

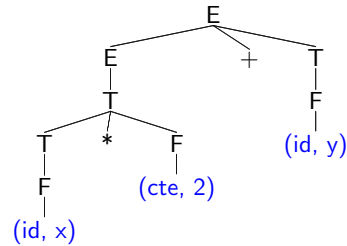
- Declaraciones de procedimientos y funciones
- Llamadas a procedimientos y funciones



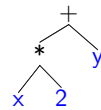
- Desarrollo de $n * m$ frente a $n + m$ compiladores.
- Descomposición inteligente de problemas.
- LI de muy bajo nivel
 - ⇒ permite una posterior *Generación de Código* sencilla y fácil de implementar
- LI independiente de la máquina
 - ⇒ permite una etapa de *Optimización Código Intermedio*, independiente de la máquina, de importancia creciente
- Parte independiente de la máquina >> parte dependiente de la máquina.

Códigos Intermedios Gráficos

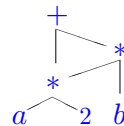
Árbol sintáctico de análisis $[x * 2 + y]$



Árbol Sintáctico Abstracto $[x * 2 + y]$



Grafos Dirigidos Acíclicos $[a * 2 + a * 2 * b]$



Códigos Intermedios Lineales

Código de una máquina a pila $[x * 2 + y]$
[por ejemplo: **bytecodes**]

```

iload x
ldc 2
imul
iload y
iadd
    
```

Código 3-direcciones $[x * 2 + y]$

```

t1 ← x
t2 ← 2
t3 ← t1 * t2
t4 ← y
t5 ← t3 + t4
    
```

CÓDIGO 3-DIRECCIONES: INVENTARIO

```

x ← y op z
x ← op z
x ← y
x ← cte
x ← pop
push x
halt
    
```

```

goto e
call e
return e
if x oprel y goto e
x ← a[i]
a[i] ← x
    
```

$x \leftarrow a[i] \quad \equiv \quad x \leftarrow *(&a + i) \quad \equiv \quad x \leftarrow *(a + i)$

$a[i] \leftarrow x \quad \equiv \quad *(&a + i) \leftarrow x \quad \equiv \quad *(a + i) \leftarrow x$

GENERACIÓN DE CÓDIGO INTERMEDIO

Expresiones e instrucciones: objetos simples

$P \Rightarrow$	$n = 0; \text{ cargaCont}(n); \Delta = 0; \Omega = 0;$
LD	
$E \Rightarrow E \text{ mod } E$	$\underline{SI} \neg [E^1.t = E^2.t = \text{tentero}] \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = \text{tentero}; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = E^1.d \text{ mod } E^2.d);$
$\Rightarrow (E)$	$E.t = E^1.t; E.d = E^1.d;$
$S \Rightarrow id = E$	$\underline{SI} \neg [\text{obtTds}(id.n, id.t, id.d) \wedge (id.t = E.t)] \{ \text{MenError}(.); \}$ $\underline{SINO} \text{Emite}(id.d = E.d);$

Ω primera instrucción libre en el *segmento de instrucciones*. **Emite**: genera una instrucción de código intermedio en la dirección Ω y posteriormente incrementa Ω . **CreaVarTemp**(t): función que crea una variable temporal para un tipo dado, devuelve el valor actual de Δ y hace $\Delta = \Delta + \text{talla}(t);$

GENERACIÓN DE CÓDIGO INTERMEDIO

Expresiones e instrucciones: objetos simples (cont.)

$E \Rightarrow id$	$\underline{SI} \neg [\text{obtTdS}(id.n, id.t, id.d) \{ \text{MenError}(.); E.t = \text{terror}; \}$ $\underline{SINO} E.t = id.t; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = id.d);$
$\Rightarrow cte$	$E.t = cte.t; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = cte.num);$
$\Rightarrow - E$	$\underline{SI} (E^1.t \neq \text{tentero}) \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = E^1.t; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = - E^1.d);$

GENERACIÓN DE CÓDIGO INTERMEDIO

Expresiones e instrucciones: objetos estructurados (registro)

$E \Rightarrow id . id$	$\underline{SI} \neg [\text{obtTdS}(id^1.n, id^1.t, id^1.d) \wedge (id^1.t = \text{registro}(id^1.lc))$ $\wedge \text{obtCampo}(id^1.lc, id^2.n, id^2.t, id^2.d)$ $] \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = id^2.t; d = id^1.d + id^2.d;$ $E.d = \text{CreaVarTemp}(E.t); \text{Emite}(E.d = d);$
$S \Rightarrow id . id = E$	$\underline{SI} \neg [\text{obtTdS}(id^1.n, id^1.t, id^1.d) \wedge (id^1.t = \text{registro}(id^1.lc))$ $\wedge \text{obtCampo}(id^1.lc, id^2.n, id^2.t, id^2.d)$ $\wedge (id^2.t = E.t)] \{ \text{MenError}(.); \}$ $\underline{SINO} d = id^1.d + id^2.d; \text{Emite}(d = E.d);$

obtCampo: función que obtiene el tipo y la posición relativa de un cierto campo, en una lista de campos de un registro. Devolverá el valor FALSE, en caso de error.

GENERACIÓN DE CÓDIGO INTERMEDIO

Expresiones e instrucciones: objetos estructurados (array)

$E \Rightarrow id [E]$	$\underline{SI} \neg [\text{obtTdS}(id.n, id.t, id.d) \wedge (id.t = \text{tvector}(id.nel, id.tel))$ $\wedge (E^1.t = \text{tentero})] \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = id.tel; \text{Emite}(E^1.d = E^1.d * \text{talla}(E.t));$ $E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = id.d [E^1.d]);$
$S \Rightarrow id [E] = E$	$\underline{SI} \neg [\text{obtTdS}(id.n, id.t, id.d) \wedge (id.t = \text{tvector}(id.nel, id.tel))$ $\wedge (E^1.t = \text{tentero}) \wedge (id.tel = E^2.t)$ $] \{ \text{MenError}(.); \}$ $\underline{SINO} \text{Emite}(E^1.d = E^1.d * \text{talla}(id.tel));$ $\text{Emite}(id.pos [E^1.pos] = E^2.pos);$

talla: función que calcula la talla asociada a un cierto tipo.

GENERACIÓN DE CÓDIGO INTERMEDIO

Expresiones lógicas

$E \Rightarrow E \text{ and } E$	$\underline{SI} \neg [(E^1, E^2).t = \text{tlógico}] \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = \text{tlógico}; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = E^1.d * E^2.d);$
$\Rightarrow E \text{ or } E$	$\underline{SI} \neg [(E^1, E^2).t = \text{tlógico}] \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = \text{tlógico}; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = E^1.d + E^2.d);$ $\text{Emite}(\text{if } E.d \leq 1 \text{ goto } \Omega + 2); \text{Emite}(E.d = 1);$
$\Rightarrow \text{not } E$	$\underline{SI} \neg [(E^1, E^2).t = \text{tlógico}] \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = \text{tlógico}; E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = 1 - E^1.d);$

GENERACIÓN DE CÓDIGO INTERMEDIO

Expresiones lógicas

$E \Rightarrow \text{true}$	$E.t = \text{tlógico}; \quad E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = '1');$
$\Rightarrow \text{false}$	$E.t = \text{tlógico}; \quad E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = '0');$
$\Rightarrow E \text{ oprel } E$	$\underline{\text{SI}} \neg[(E^1, E^2).t \in \{\text{tentero}, \text{treal}\}]$ $\{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{\text{SINO}} \quad E.t = \text{tlógico}; \quad E.d = \text{CreaVarTemp}(E.t);$ $\text{Emite}(E.d = '1');$ $\text{Emite}(\text{if } E^1.d \text{ oprel } E^2.d \text{ goto } \Omega + 2);$ $\text{Emite}(E.d = '0');$

LISTAS DE REFERENCIAS NO SATISFECHAS

Segmento de código

L.A.N.S

\vdots		\vdots
i: goto \otimes	$\Leftarrow A = \text{CreaLans}(i)$	A i
\vdots		\vdots
j: if \dots goto \otimes	$\Leftarrow B = \text{CreaLans}(j)$	B j
\vdots		\vdots
k: \dots	$\Leftarrow C = \text{Fusionalans}(A, B)$	

LISTAS DE REFERENCIAS NO SATISFECHAS

Segmento de código

L.A.N.S

\vdots		\vdots
i: goto \otimes	$\Leftarrow A = \text{CreaLans}(i)$	C i, j
\vdots		\vdots
j: if \dots goto \otimes	$\Leftarrow B = \text{CreaLans}(j)$	
\vdots		
k: \dots	$\Leftarrow C = \text{Fusionalans}(A, B)$	
\vdots		
n: \dots	$\Leftarrow \text{CompletaLans}(C, n)$	

GENERACIÓN DE CÓDIGO INTERMEDIO

Instrucciones que implican rotura del flujo de control

$S \Rightarrow \text{if}(E)$	$\underline{\text{SI}}(E.t \neq \text{tlógico}) \{ \text{MenError}(.); \}$ $S.lf = \text{CreaLans}(\Omega); \quad \text{Emite}(\text{if } E.d = '0' \text{ goto } \otimes);$ $S.fin = \text{CreaLans}(\Omega); \quad \text{Emite}(\text{goto } \otimes); \quad \text{CompletaLans}(S.lf, \Omega);$ $\text{CompletaLans}(S.fin, \Omega);$
$\Rightarrow \text{while}(E)$	$S.ini = \Omega;$ $\underline{\text{SI}}(E.t \neq \text{tlógico}) \{ \text{MenError}(.); \}$ $S.lf = \text{CreaLans}(\Omega); \quad \text{Emite}(\text{if } E.d = '0' \text{ goto } \otimes);$ $S \quad \text{Emite}(\text{goto } S.ini); \quad \text{CompletaLans}(S.lf, \Omega);$

CreaLans: función que crea una lista de argumentos no satisfechos.

CompletaLans: completa una lista de argumentos no satisfechos.

GENERACIÓN DE CÓDIGO INTERMEDIO

Instrucciones que implican rotura del flujo de control (cont.)

$S \Rightarrow \text{do}$	$S.\text{ini} = \Omega;$
$S \text{ while } (E)$	$\underline{SI} (E.t \neq \text{tlógico}) \{ \text{MenError}(.); \}$ $\text{Emite}(\text{if } E.d = '1' \text{ goto } S.\text{ini});$
$\Rightarrow \text{for } (E ;$	$S.\text{ini} = \Omega;$
$E ;$	$\underline{SI} (E_2.t \neq \text{tlógico}) \{ \text{MenError}(.); \}$ $S.\text{lv} = \text{CreaLans}(\Omega); \text{Emite}(\text{if } E_2.d = '1' \text{ goto } \otimes);$ $S.\text{lf} = \text{CreaLans}(\Omega); \text{Emite}(\text{goto } \otimes);$ $S.\text{aux} = \Omega;$
$E)$	$\text{Emite}(\text{goto } S.\text{ini}); \text{CompletaLans}(S.\text{lv}, \Omega);$
S	$\text{Emite}(\text{goto } S.\text{aux}); \text{CompletaLans}(S.\text{lf}, \Omega);$

GENERACIÓN DE CÓDIGO INTERMEDIO

Llamadas a funciones

$E \Rightarrow \text{id } ($	$\underline{SI} \neg [\text{obtTdS}(\text{id}.\text{nom}, \text{id}.\text{t}, \text{id}.\text{dpi})$ $\wedge (\text{id}.\text{t} = \text{tfunción}(\text{id}.\text{td}, \text{id}.\text{tr}, \text{id}.\text{tasp}))]$ $\{ E.t = \text{terror}; \text{MenError}(.); \}$ $\underline{SINO} E.t = \text{id}.\text{tr};$
$A)$	$\text{Emite}(\text{sp} = \text{sp} + \text{talla}(E.t));$ $\underline{SI} (A.t \neq \text{id}.\text{dom}) \{ E.t = \text{terror}; \text{MenError}(.); \}$ $\text{Emite}(\text{push}(\Omega + 2)); \text{Emite}(\text{call id}.\text{dpi}); \text{Emite}(\text{sp} = \text{sp} - \text{id}.\text{tsp});$ $E.d = \text{CreaVarTemp}(E.t); \text{Emite}(E.d = \text{pop});$
$A \Rightarrow \epsilon$	$A.t = \text{tvacio};$
$\Rightarrow \text{LA}$	$A.t = \text{LA}.\text{t};$
$\text{LA} \Rightarrow E$	$\text{LA}.\text{t} = E.t; \text{Emite}(\text{push}(E.d));$
$\Rightarrow E ,$	$\text{Emite}(\text{push}(E.d));$
LA	$\text{LA}.\text{t} = E.t \otimes \text{LA}.\text{t};$

GENERACIÓN DE CÓDIGO INTERMEDIO

Funciones y parámetros

$D \Rightarrow$	$n++;$ cargaCont (n); $D.\text{aux} = \Delta;$ $\Delta = 0;$
$T \text{ id } (PF)$	insTdS (id.nom, "función", tfunción(PF.t, T.t, PF.talla), n-1, Ω); Emite (push(fp)); Emite (fp = sp); $D.d = \text{CreaLans}(\Omega); \text{Emite}(\text{sp} = \text{sp} + \otimes);$ $\{ DL LI \}$ CompletaLans (D.d, Δ); Emite (sp = fp); Emite (fp = pop); Emite (return(pop)); descargaCont (n); $n--;$ $\Delta = D.\text{aux};$
$PF \Rightarrow \epsilon$	$PF.t = \text{tvacio}; PF.talla = 0;$
$\Rightarrow \text{LF}$	$PF.t = \text{LF}.\text{t}; PF.talla = \text{LF}.\text{talla} - \text{TallaSegEnlaces};$
$\text{LF} \Rightarrow \text{DV}, \text{LF}$	$\text{LF}.\text{t} = \text{DV}.\text{t} \otimes \text{LF}.\text{t}; \text{LF}.\text{talla} = \text{LF}.\text{talla} + \text{DV}.\text{talla};$ insTdS (DV.nom, "parámetro", DV.t, n, -LF.talla);
$\Rightarrow \text{DV}$	$\text{LF}.\text{t} = \text{DV}.\text{t}; \text{LF}.\text{talla} = \text{TallaSegEnlaces} + \text{DV}.\text{talla};$ insTdS (DV.nom, "parámetro", DV.t, n, -LF.talla);

EJEMPLO-1

$S \Rightarrow S ; S$	$S.b = \text{FusionaLans}(S^1.b, S^2.b);$
$\Rightarrow \text{switch } (E) \{$	$\underline{SI} (E.t \neq \text{tentero}) \text{MenError}(.);$
$L \}$	$L.d = E.d; L.h = \text{nil};$
$\Rightarrow \text{break}$	CompletaLans (L.b, Ω); $S.b = \text{nil};$
$L \Rightarrow \text{case cte} :$	$\underline{SI} (\text{cte}.\text{t} \neq \text{tentero}) \text{MenError}(.);$
S	$L.\text{fin} = \text{CreaLans}(\Omega); \text{Emite}(\text{if } \text{cte}.\text{num} \neq L.d \text{ goto } \otimes);$ CompletaLans (L.h, Ω);
L	$L^1.h = \text{CreaLans}(\Omega); \text{Emite}(\text{goto } \otimes);$
$\Rightarrow \epsilon$	$L^1.d = L.d; \text{CompletaLans}(L.\text{fin}, \Omega);$
$\Rightarrow \text{default} :$	$L.b = \text{FusionaLans}(S.b, L^1.b);$
S	CompletaLans (L.h, Ω);
	$L.b = S.b;$