

Regole di scoping

servono se esistono riferimenti non locali nelle procedure

Statico

Risoluzione dei riferimenti non locali avviene nell'ambiente di definizione delle procedure

Dinamico

Risoluzione dei riferimenti non locali avviene nell'ambiente di esecuzione delle procedure

IMPLEMENTAZIONE

→ Display

→ Catena statica

(catena dinamica)
→ A-list

→ CRT (tabella controllo dei riferimenti)

IMPLEMENTAZIONE Scoping statico mediante CATENA STATICA

→ Costruzione della catena dei link statici

→ L'utilizzo efficiente della catena per risolvere i riferimenti

→ Primo passo → si alloca l'RTA sullo stack

Link dinamico
(link al chiamante)
è l'RTA precedente nello stack

Link statico
link al genitore statico, ovvero all'RTA che contiene il testo della procedura

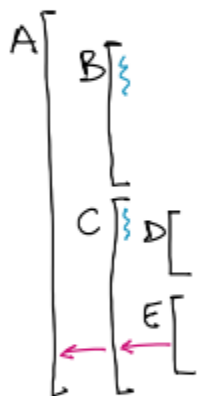
CATENA STATICA → catena dei link/puntatori statici

LINK STATICO → è il puntatore al genitore statico, ovvero all' RDA del blocco che contiene il testo della definizione della procedura

PROFONDITÀ STATICA (Sd) → è il valore della profondità di annidamento della definizione della procedura

Esempio: Sequenza di chiamate $\underline{A} \xrightarrow{\text{chiamata}} B \xrightarrow{\text{chiamata}} C \xrightarrow{\dots} D \rightarrow E \rightarrow C$

Sequenza di annidamento statico



B e C sono definiti nel testo dentro A

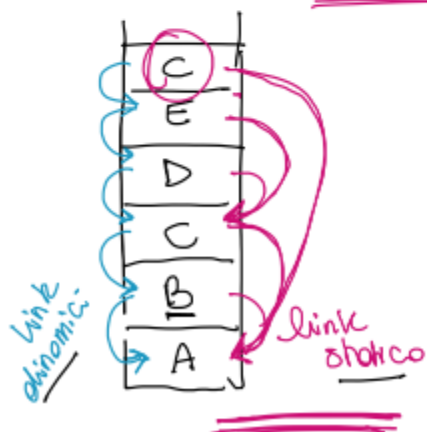
D e E sono definiti nel testo dentro C

$$Sd(A) = 0$$

$$Sd(B) = Sd(C) = 1$$

$$(Sd(\text{Blocco}) = Sd(\text{Blocco che contiene}) + 1)$$

$$Sd(D) = Sd(E) = 2$$



Chiamata A → P → B

Dobbiamo costruire al momento della chiamata il giusto link statico

→ è determinato dalla procedura chiamante

e dipende dalla profondità statica dei blocchi coinvolti

In generale : Ch procedura chiamante
P procedura chiamata

Vogliamo calcolare K che corrisponde al numero di volte in cui dobbiamo risalire la catena statica per trovare il genitore statico. (a partire da Ch)

\Rightarrow Se P è definito dentro Ch (se Ch è il genitore statico di P)
 $\Rightarrow k=0$

Altrimenti $k = \text{Sdl}(\text{Ch}) - \text{Sdl}(P) + 1$

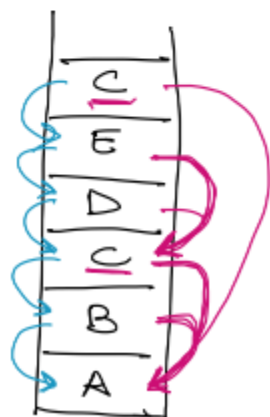
\hookrightarrow Se $k=0$ allora il link statico punta al chiamante

\hookrightarrow Se $k > 0$ allora risalire la catena statica da Ch k volte, il blocco raggiunto è il target del link statico

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow C$



$\text{Sdl}(A) = 0$
 $\text{Sdl}(B) = \text{Sdl}(C) = 1$
 $\text{Sdl}(D) = \text{Sdl}(E) = 2$



$$k_B = \text{Sdl}(A) - \text{Sdl}(B) + 1$$

$$= 0 - 1 + 1 = 0 \Rightarrow \text{LS}(B) = A$$

chiamato

chiamante

$$K_C = Sd(B) - Sd(C) + 1 = 1 - 1 + 1 = \underline{1} \Rightarrow \underline{Ls(C)} = \underline{Ls(B)} = \underline{A}$$

chiamato chiamato

$$K_D = Sd(C) - Sd(D) + 1 = 1 - 2 + 1 = 0 \Rightarrow Ls(D) = C$$

$$K_E = Sd(D) - Sd(E) + 1 = 2 - 2 + 1 = \underline{1} \Rightarrow \underline{Ls(E)} = \underline{Ls(D)} = \underline{C}$$

chiamato chiamato

$$K_C = Sd(E) - Sd(C) + 1 = 2 - 1 + 1 = \underline{2} \Rightarrow \underline{Ls(C)} = \underline{Ls(Ls(E))} = \underline{Ls(C)} = \underline{A}$$

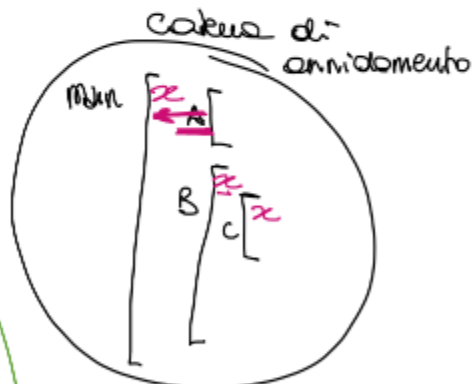
ESEMPLO

```
int x
void A() { x = x + 1; } A
```

```
void B() { int x;
```

```
void C(int y) {
  int x;
  x = y + 2;
  A();
}
```

```
x = 0;
A();
C(3);
}
```



$Sd(main) = 0$

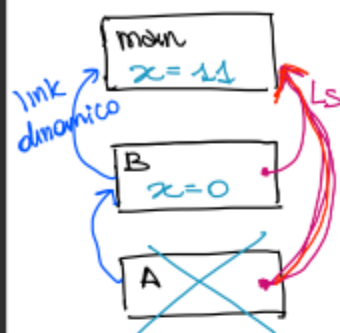
$Sd(A) = Sd(B) = 1$

$Sd(C) = 2$

main → B → A
 ↓
 C → A

Sequenza

B();
↳ ↖



$$K_B = Sd(main) - Sd(B) + 1$$

$$= 0 - 1 + 1 = 0$$

$$Ls(B) = main$$

$$K_A = Sd(B) - Sd(A) + 1$$

$$= 1 - 1 + 1 = 1$$

$$Ls(A) = Ls(B)$$

$$= main$$

Risoluzione di x :

$$N_A = Sd(A) - Sd(main)$$

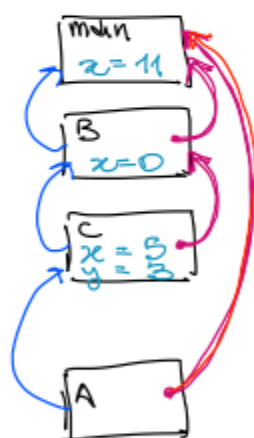
$$= 1 - 0 = 1$$

→ risolviamo di 1
o meno stocche e
troviamo x

Chiamate di C(B)

Come troviamo la x
modificata da A

↳ dobbiamo calcolare di punto
risolvere la catena stocica per
trovare x



$$K_C = Sd(B) - Sd(C) + 1$$

$$= 1 - 2 + 1 = 0$$

$$Ls(C) = B$$

$$K_A = Sd(C) - Sd(A) + 1$$

$$= 2 - 1 + 1 = 2$$

$$Ls(A) = Ls(Ls(C)) = Ls(B) = main$$

↳ dobbiamo calcolare di punto risolvere
la catena stocica per trovare la x
di riferimento per A

$$\text{Risoluzione di } x: N_A = Sd(A) - Sd(main) = 1$$

Risoluzione di riferimenti

In generale : supponiamo che x (non locale) sia usato in P . Supponiamo che D sia il blocco più vicino nella catena di annidamento che definisce x .

$$N = Sd(P) - Sd(D)$$



procedura
che usa x

la più vicina
procedura (nella catena
di annidamento)
che definisce x

è il numero di volte che
devo visitare la catena statica da P per
trovare il blocco che contiene la x di riferimento