Esercizi e Problemi

Sommario: 26 Aprile, 2021

• Trassmissione: Deep e Shallow Binding in Static Scope

• Eccezioni: Controllo di eccezioni in Small21L2.ml

Tipi: Un sistema di tipi sound per Small21

Esercizio: Trasmissione di funzione

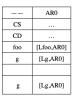
Consideriamo il programma sotto, in linguaggi a blocchi con scope statico ma:

- (a) Deep Binding,
- (b) Shallow Binding rispettivamente.

```
{void foo (int f(), int x){
    int fie(){
        return x,
    }
    int z,
    if (x==0) z=f(),
    else foo(fie,0),
}
int g(){
    return 1,
    }
foo(g,1),
}
```

Si mostri e si commenti la sequenza di stati attesa dalla computazione di esecutori dei 2 differenti tipi di Linguaggio.

DEEP BINDING IN STATIC SCOPE





SHALLOW BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]
g	g

Lfoo	foo-Code
Lg	g-Code
MEN	ORIA

Sopra i differenti stati durante la trasmissione dell'invocazione foo(g,1).

• Deep Binding: Il valore calcolato per il parametro g è la denotazione di g in AR0.

void foo(int f(), int x){

return x;

else foo(fie.0):

int fie(){

int z; if (x == 0) z = f(x):

int g(){
 return 1;
}
foo(g,1);

• Shallow Binding: Il valore calcolato per il parametro g è l'identitificatore g.

DEEP BINDING IN STATIC SCOPE



Lfoo	foo-Code
Lg	g-Code
Lfie	fie-Code
Lx	1
Lz	
MEN	MORIA .

{
<pre>void foo(int f(), int x){</pre>
int fie(){
return x;
}
int z;
if $(x == 0) z = f(x);$
else foo(fie,0);
}
int g(){
return 1;
}
foo(g,1);
}

SHALLOW BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]
g	g
	ARfoo
CS	
CD	
f	g
x	Lx
fie	[Lfie,ARfoo]
z	Lz

Lfoo	foo-Code
Lg	g-Code
Life	fie-Code
Lx	1
Lz	
	1

Sopra i differenti stati durante l'invocazione di foo(g,1): Prima della valutazione del condizionale.

- Deep Binding: il parametro f ha come denotazione la chiusura di g in ARO.
- Shallow Binding: Il parametro f ha come denotazione <u>l'identitificatore</u> g.

DEEP BINDING IN STATIC SCOPE

	AR0	
CS		
CD		
foo	[Lfoo,AR0]	
g	[Lg,AR0]	
g	[Lg,AR0]	
	ARfoo	
CS		
CD		
f	[Lg,AR0]	
x	Lx	
fie	[Lfie,ARfoo]	
z	Lz	

false [Lfie.ARfoo]

x == 0

Lfoo foo-Code Lg g-Code Lfie fie-Code Lx 1 Lz	g-Code
Lfie fie-Code Lx 1 Lz	
Lx 1	fie-Code
Lz	
	1
MEMODIA	
	MORIA
14121	

{	
ì	oid foo(int f(), int x){
	int fie(){
	return x;
	}
	int z;
	if $(x == 0) z = f(x);$
	else foo(fie,0);
	}
i	nt g(){
	return 1;
}	
f	oo(g,1);
}	

SHALLOW BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]
g	g
	ARfoo
CS	
CD	
f	g
х	Lx
fie	[Lfie,ARfoo]
z	Lz
x == 0	false
fie	fie

Lfoo	foo-Code
Lg	g-Code
Lx	1
Lz	

Sopra i differenti stati durante l'invocazione di foo(g,1): Valutazione del condizionale e prima dell'invocazione ricorsiva foo(fie,0).

- Deep Binding: Il valore calcolato per il parametro fie è la denotazione di fie in ARfoo.
- Shallow Binding: Il valore calcolato per il parametro fie è l'identitificatore fie

DEEP BINDING IN STATIC SCOPE

	AR0		Lfoo	foo-Co
CS			Lg	g-Coc
CD			Lfie	fie-Co
foo	[Lfoo,AR0]		Lx	- 1
g	[Lg,AR0]		Lz	
	Lagrana		Lx1	0
			Lz1	
g	[Lg,AR0]	li		
	ARfoo			
CS				
CD				
f	[Lg,AR0]			
x	Lx			
fie	[Lfie,ARfoo]		MEN	IORIA
z	Lz	'		
x == 0	false			
fie	[Lfie,ARfoo]			
	ARfoo1			
CS				

{...
void foo(int f(), int x){
 int fie() {
 return x;
 }
 int z;
 int z;
 if (x == 0) z = f(x);
 clse foo(fie,0);
 }
 int g() {
 return 1;
 }
 foo(g,1);

SHALLOW BINDING IN STATIC SCOPE

эпаі	TOM BINDIN	GINSI	A11	CSCOPE
	AR0	Lfo	0	foo-Code
CS		Lg	П	g-Code
CD		Lfie		fie-Code
foo	[Lfoo,AR0]	Lx	П	1
g	[Lg,AR0]	Lz		
	(Ligatio)	Lx	ı	0
		Lzi	П	
g	g		T	
	ARfoo			
CS				
CD				
f	8			
x	Lx			
fie	[Lfie,ARfoo]			
z	Lz	_	_	
x == 0 fie	false fie			
	ARfool			
cs				
CD				
f	fie			
x	Lxl			
fie	[Lfie,ARfoo1]			
z	Lzl			
x == 0	true			
f(x) =	0			
х.	"			

Sopra i differenti stati durante l'invocazione di foo(g,1): Invocazione ricorsiva foo(fie,0) - Valuatazione del condizionale e trasmissione parametri dell'invocazione f(x).

- Deep Binding: La funzione f da invocare è la denotazione di f, ovvero la chiusura di fie in ARfoo.
- Shallow Binding: La funzione f da invocare è la funzione legata all'identificatore fie in ARfoo1, e questa è la denotazione di fie in ARfoo1, ovvero la chiusura di fie in ARfoo1.

CD ...

f [Lfie,ARfoo]

x Lx1

fie [Lfie,ARfool

z Lz1

x == 0 f(x) = x 0

DE	EP BINDING IN	STATIC	SCOPE		SHA	LLOW BINDIN	G IN STAT	IC SCOPE
	AR0	Lfoo	foo-Code	{		AR0	Lfoo	foo-Code
CS		Lg	g-Code	void foo(int f(), int x){	CS		Lg	g-Code
CD		Life	fie-Code	int fie(){	CD		Life	fie-Code
foo	[Lfoo,AR0]	Lx	1	return x;	foo	ILfoo,AR01	Lx	- 1
8	[Lg,AR0]	Lz		}		[Lg,AR0]	Lz	
- 8	[Lg,AKO]	Lx1	0	int z;	g	[Lg,AK0]	Lx1	0
		Lzl		if $(x == 0) z = f(x)$;			Lzl	
8	[Lg,AR0]			else foo(fie,0);	2	2		
	ARfoo			}		ARfoo		
CS				int g(){	CS			
CD				return 1;	CD			
f	[Lg,AR0])	X	Lx		
х	Lx			foo(g,1);	fle	[Lfie,ARfoo]		
fie	[Lfie,ARfoo]	ME	MORIA	}	2	Lz		
z	Lz		- Indicate					
					x == 0	false		
x == 0 fie	false [Ltie,ARfoo]				fie	fie		
	ARfool					ARfool		
					CS			
CD					CD			
f	[Lfie,ARfoo]				f	fie		
x	Lxl				x	Lx1		
fie	[Lfie,ARfool				fie	[Lfle,ARfoo1]		
z	Lzl				z	Lz1		
					-			
x == 0	true				x == 0 f(x) =	true		
f(x) =	-	_			x	0 1		
х	0	1				ARtie		
	ARfie	1			CS	ARfool		
CS	ARfoo	1			CD			
RA	7 -				RA		_	
- KA								
x	1				x	0		

Sopra i differenti stati durante l'invocazione di foo(g,1): Invocazione ricorsiva foo(fie,0) - invocazione di f(x), dove $f \` e$ un parametro funzione.

- Deep Binding: Invocazione di fie in ARfoo.
- Shallow Binding: Invocazione di fie in ARfoo1.

DEEP BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]
g	[Lg,AR0]
	ARfoo
CS	
CD	
f	[Lg,AR0]
x	Lx
fie	[Lfie,ARfoo]
z	Lz
x == 0	false
fie	[Lfie,ARfoo]
	ARfoo1
CS	
CD	
f	[Lfie,ARfoo]
x	Lx1
fie	[Lfie,ARfoo1]
z	Lzl
x == 0	true
f(x) =	1
X	0

Lfoo	foo-Code
Lg	g-Code
Life	fie-Code
Lx	1
Lz	
Lx1	0
Lzl	1

MEMORIA

	{
	void foo(int f(), int x){
	int fie(){
	return x;
1	}
1	int z;
1	if $(x == 0) z = f(x);$
1	else foo(fie,0);
	}
	int g(){
	return 1;
	}
	foo(g,1);
	}

SHALLOW BINDING IN STATIC SCOPE

	AR0	
CS		
CD		
foo	[Lfoo,AR0]	
g	[Lg,AR0]	
g	g	
	ARfoo	
CS		
CD		
f	g	
x	Lx	
fie	[Lfie,ARfoo]	
z	Lz	
x == 0	false	
fie	fie	
	ARfool	
CS		
CD		
f	fie	
x	Lx1	
fie	[Lfie,ARfoo1]	
z	Lz1	
x == 0	true	
f(x) =	0	
x	0	

1	Lfoo	foo-Code
1	Lg	g-Code
1	Life	fie-Code
1	Lx	1
1	Lz	
ł	Lx1	0
ł	Lzl	0
4		
+		
1		
1		
1		
1		
1		
Ĭ		
1		

Sopra i differenti stati durante l'invocazione di foo(g,1): Invocazione ricorsiva foo(fie,0) - assegnamento z = f(x).

- Deep Binding: idem.
- Shallow Binding: idem.

DEEP BINDING IN STATIC SCOPE



foo-Code
g-Code
fie-Code
1
MORIA

```
{...
  void foo(int f(), int x){
    int fie(){
      return x;
  }
  int z;
  if (x == 0) z = f(x);
  else foo(fie,0);
  }
  int g(){
  return 1;
  }
  foo(g,1);
}
```

SHALLOW BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]
g	g
	ARfoo
CS	
CD	
f	g
x	Lx
fie	[Lfie,ARfoo]
z	Lz
x == 0	false
fie	fie

Lfoo	foo-Code
Lg	g-Code
Life	fie-Code
Lx	1
Lz	

Sopra i differenti stati durante l'invocazione di foo(g,1): Invocazione ricorsiva foo(fie,0) - ritorno.

- Deep Binding: idem.
- Shallow Binding: idem.

DEEP BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]



{... void foo(int f(), int x){ int fie(){ return x; } int z; if (x == 0) z = f(x); else foo(fie,0); } int g(){ return 1; } foo(g,1); }

SHALLOW BINDING IN STATIC SCOPE

	AR0
CS	
CD	
foo	[Lfoo,AR0]
g	[Lg,AR0]

Lfoo	foo-Code
Lg	g-Code

Sopra i differenti stati durante l'invocazione di foo(g,1): Ritorno dall'invocazione e Stato Finale.

- Deep Binding: Computazioni diverse.
- Shallow Binding: Computazioni diverse.
- Coppia (Stato-Iniziale, Stato-Finale) identica: Calcolano la Stessa Funzione Calcolabile?

Esercizio: Controllo di Eccezioni nel codice di Small21L2

Nel materiale distribuito oggi dalla pagina del corso, alla voce Soluzione.ml troviamo il listing con l'ultimo aggiornamento sul Laboratorio. Il listing contiene il codice per risolvere gli esercizi 7 ed 8. L'esercizio 9 mostra alcuni problemi sul dump di ambiente, in particolare il dump conduce al sollevamento di un' eccezione. Si risponda nell'ordine:

- (a) Quale eccezione è sollevata e perchè?
- (b) Si fornisca una modifica al codice in modo da intercettare tale eccezione e trattarla al fine di ottenere (la) una presentazione (richiesta) per l'ambiente

Esercizio: Un sistema di tipi sound per Small21

Oggi applichiamo a Small21 quanto detto nella Lezione9 sui Tipi e sui sistemi di Tipi per garantire i programmi contro la generazione di stati di "stuck". Nel materiale di oggi, l'allegato SistemaYSmall21 contiene le regole di inferenza di un sistema deduttivo per il controllo di tipi di Small21. Il sistema deve essere completato. Allo scopo:

- (a) Considerare e commentare le regole date
- (b) Procedere con la definizione delle regole per le seguenti strutture:
 - Dichiarazione array monodimensione, modificabile solo nei componenti (oggi);
 - Dichiarazione procedura/funzione con al più un parametro per valore o per reference (oggi);
 - Completare le regole per le espressioni di Small21
 - Completare le regole per i comandi di Small21
 - Completare le regole per gli statements di Small21
 - Completare le regole per il programma di Small21