

Simulazione de

(1)
Sim F

```

0: public class Fib {
1:   public static int fib(int n) {
2:     if (n==0 || n==1) {
3:       return 1;
4:     } else {
5:       return fib(n-1) + fib(n-2);
6:     }
7:   }
8:   public static void main (String[] args) {
9:     int v = fib(3);
10:  }

```

program counter	frame stack
0	"Voto"
5	<div> <div> v ? args — return? result? </div> main </div>
6 (È necessario ridimensionare f)	<div> <div> v ? args — 6 ? </div> main </div>
1	<div> <div> M-3 ? ? v ? args — 6 ? </div> f main </div>

program counter

frame stack

(2)
SIM

2

(la volutazione
a false di
 $m=0 \parallel m=1$
ha procedere
verso 4)

configurazione
identica alla
precedente.

4

(è memoria
richiamare
(la prima osservazione di)
L)

M 3	
4	?
5	?
ops	/
6	?

L

main

1

M 2	
?	?
M 3	
4	?
5	?
ops	/
6	?

L

L

main

2

(la volutazione a
false di
 $m=0 \parallel m=1$
ha procedere verso
4)

configurazione
identica alla
precedente

program counter

frame stack

4
(è necessario
richiamare (la
prima occorrenza
di) L)

M 2		
4		?
M 3		
4		?
5 ?		
args 1		
6		?

L
 L
main

1

M 1		
?		?
M 2		
4		?
M 3		
4		?
5 ?		
args -		
6		?

L
 L
 L
main

2
(Il predicato
 $M == 0 \parallel M == 1$
è vero)

configurazione
valutiva

3

(return memorizza
2o il risultato
nell'opposite
zone nel frame
del chiamante
e riprende dal
program counter
di quest'ultimo)

M 2		
4		1
M 3		
4		?
5 ?		
args -		
6		?

L
 L
main

program counter

frame stack

(2)
5/14

4

(è necessario richia-
mare la seconda
occorrenza di L)

configurazione
identica

1

M 0		
?		?
M 2		
4		1
M 3		
4		?
5?		
args -		
6		?

L

L

L

main

2
(il predicato
 $n=0 \parallel m=1$
è vero)

configurazione
identica

3

(return permette
la memorizzazione
del corretto risultato
nell'apporto spazio
del chiamante e
rimozione del
program counter di
quest'ultimo)

M 2		
4		1+1
M 3		
4		?
5?		
args -		
6		?

L

L

main

program counter

frame stack

4

(abbiamo
terminato la
interpretazione
di $f(2)$ ne servono
per calcolare $f(3)$)

M 3		
4	1+1	f
v ? args -		
6	?	main

4

(serve calcolare
il risultato
della sequenza
chiamata di f ,
in $f(3)$)

configurazione
identica

1

M 1		
?	?	f
M 3		
4	1+1	f
v ? args -		
6	?	main

2

(il predicato
 $m == 0 \parallel m == 1$
permette l'esecuzione
di 3)

configurazione
identica

program counter

frame stack

3

(return permette
la memorizzazione
del risultato nel
posto corretto del
chiamante,
riprendendolo da
dove si era interrotto)

M 3		f
4	1+1+1	
v ? args /		main
6	?	

4
(interpretiamo
return restituendo
il risultato
al chiamante
siccome $f(2)$ e $f(1)$
nella definizione
di $f(3)$ non stati
collocati)

v ? args /		main
6	3	

6

(si conclude
l'esecuzione)

v 3 args /	
6	3