

SSIS 2007

Maria Rita Laganà, Dipartimento di Informatica
Università di Pisa

Incontro II:

Ambienti di programmazione multi-
agente
la ricorsione e il parallelismo

Esempi ricorsivi

- Un albero binario è
 - un nodo
 - una coppia di due sottoalberi binari che si dipartono da un nodo

```
to inizia
```

```
ca
```

```
crt 1
```

```
ask-turtles[pd]
```

```
end
```

```
inizia ramo 10 15
```

```
to ramo :ricorsione :lun
```

```
if :ricorsione = 0 [stop]
```

```
fd :lun rt 20
```

```
ramo :ricorsione - 1 :lun lt
```

```
40
```

```
ramo :ricorsione - 1 :lun rt 20
```

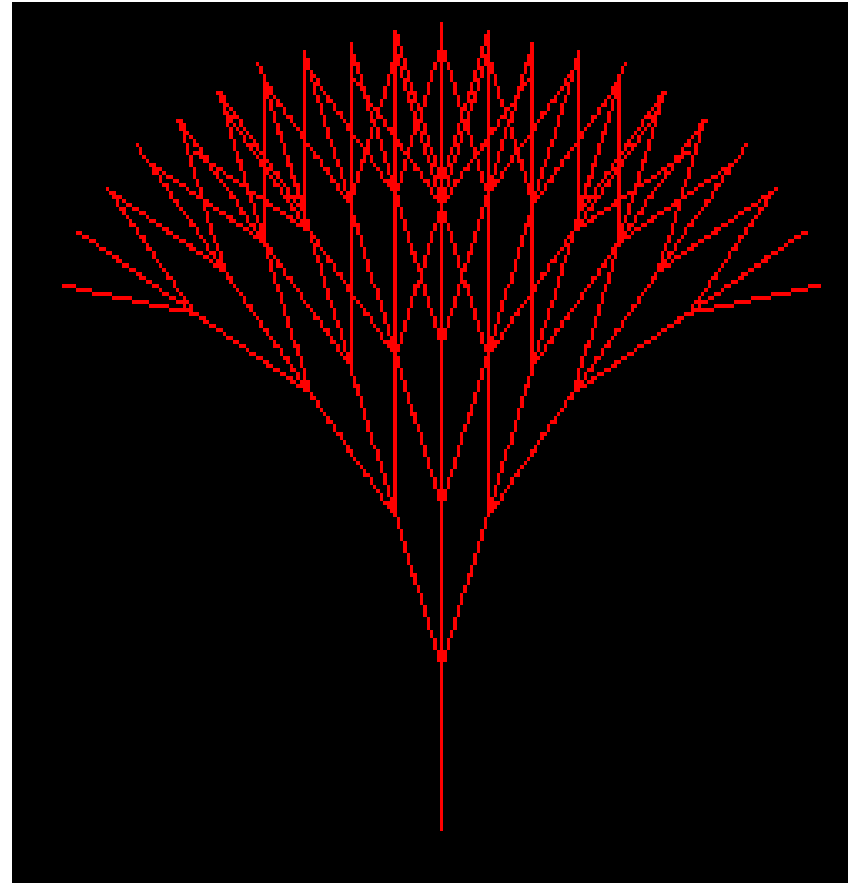
```
bk :lun ;allo stato iniziale
```

```
end
```

L'albero in parallelo

Osservare a destra
cammina 40
ripetuto 4 volte

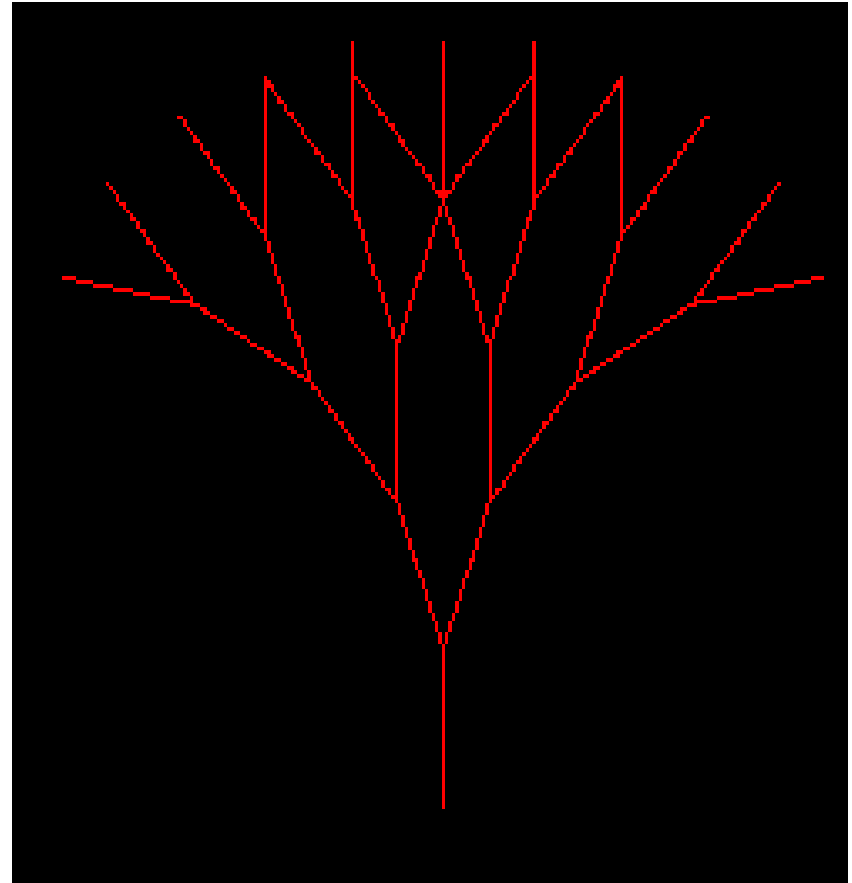
```
to cammina :lun  
fd :lun  
hatch[rt 20 ]  
hatch[lr 20 ]  
end
```



L'albero in parallelo

Osservare a destra
cammina 40
ripetuto 5 volte

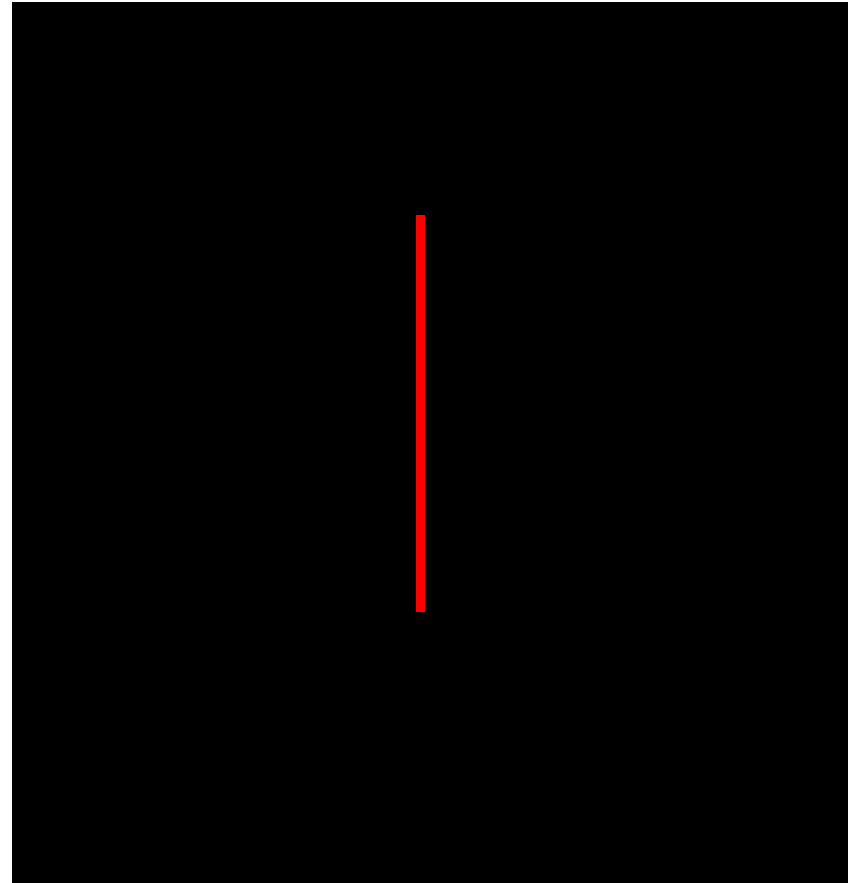
```
to cammina :lun  
fd :lun  
hatch[rt 20 ]  
hatch[lr 20 ]  
die  
end
```



Sorpresa?

Osservare a destra
repeat 5 [cammina 40]

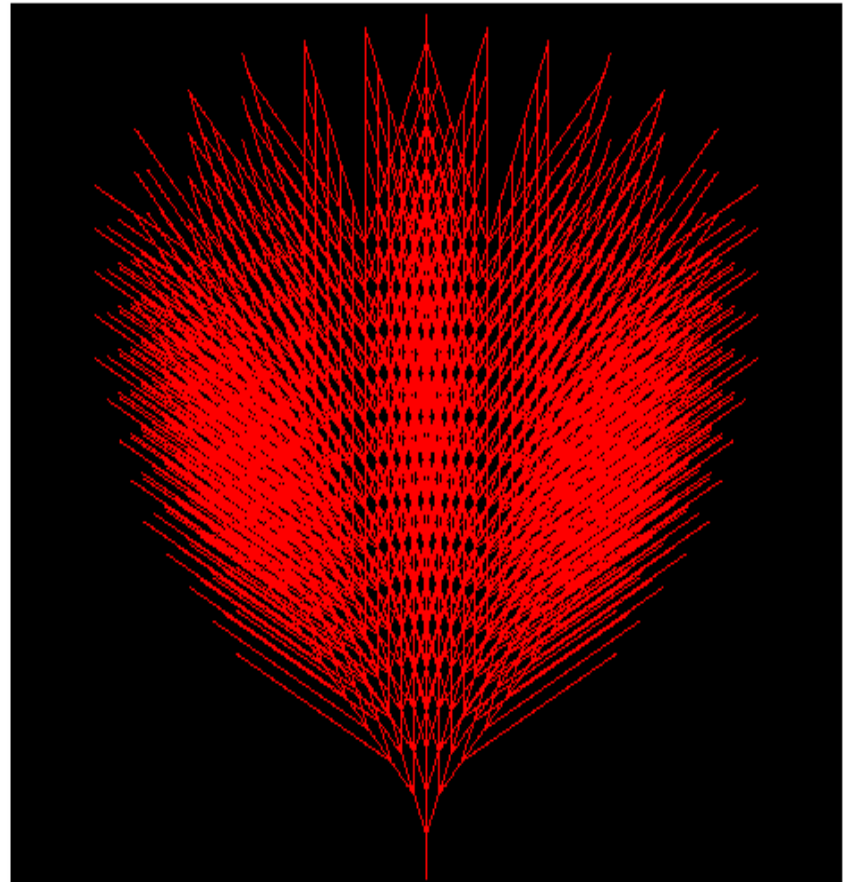
```
to cammina :lun  
  fd :lun  
  hatch[rt 20 ]  
  hatch[lr 20 ]  
die  
end
```



Sorpresa?

Osservare a destra
bk 200
repeat 5 [cammina 20]
ripetuto 4 volte

```
to cammina :lun  
fd :lun  
hatch[rt 20 ]  
hatch[lr 20 ]  
end
```



Spiegazione

- L'ordine viene eseguito dalle tartarughe attive in quel momento e non dalle tartarughe figlie.
- Due soluzioni
 - bottone forever
 - sulla finestra osservatore
 - repeat 5 [ask-turtles [cammina 20]]
- In generale è molto interessante scrivere i due algoritmi, sequenziale e parallelo.

osservazioni

```
to inizia  
ca  
crt 1  
ask-turtles [set generazione 0]  
end
```

```
turtles-own [generazione]
```

```
to genera  
hatch [ ] fd 1  
end
```

Differenza tra

usare 4 volte la procedura genera

```
repeat 4 [genera]
```

Quante sono le tartarughe? Come si dispongono?

osservazioni

```
repeat 10 [genera ]          turtles-own [generazione]
rt 90
repeat 10 [genera ]          to inizia
rt 90                          ca crt 1
                                end
```

```
repeat 2 [repeat 10 [genera ] rt 90]
repeat 2 [repeat 5 [genera ] rt 90]
repeat 4 [repeat 5 [genera ] rt 90]
```

Quante sono le tartarughe? Come si dispongono?

Un tassello e una bandiera

```
repeat 10 [ genera setc color + 10 ]  
rt 90  
repeat 10 [ genera setc color + 10 ]  
rt 90  
repeat 10 [ genera setc color + 10 ]  
rt 90  
repeat 10 [ genera setc color + 10 ]
```

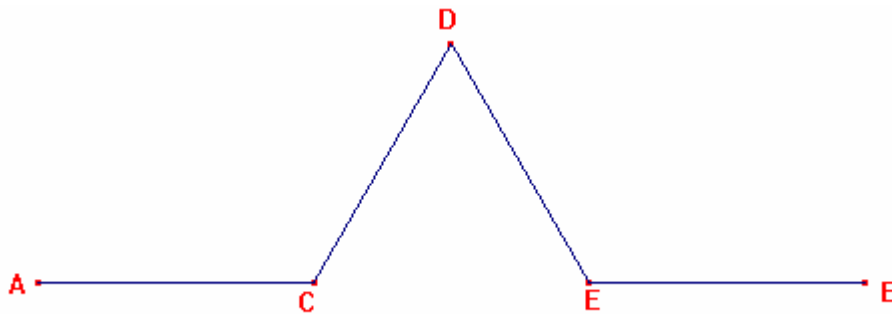
Un tassello

```
repeat 10 [ genera setc color + 10 ]  
rt 90  
repeat 20 [genera]
```

Una bandiera

Curva di Kock

- Definizione ricorsiva
 - un segmento
 - -quattro segmenti (della stessa struttura della curva di Kock) ottenuti dividendo in tre parti uguali il segmento e sostituendo la parte centrale con un triangolo equilatero senza base.



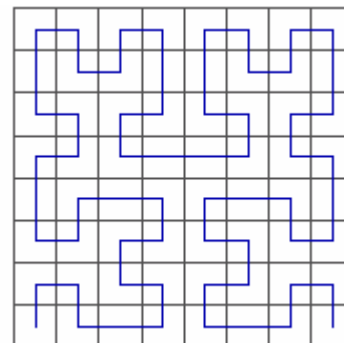
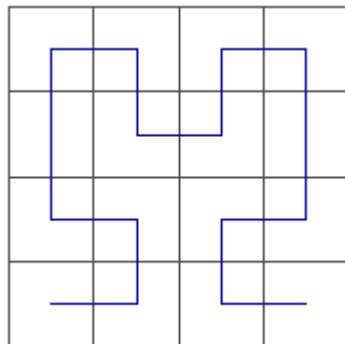
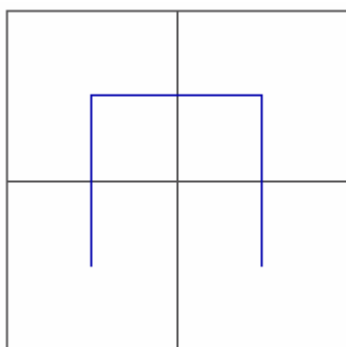
Kock ricorsivo

```
to kock :dim :prof
if :prof = 0 [fd :dim stop]
kock :dim / 3 :prof - 1
lt 60 kock :dim / 3 :prof - 1
rt 120 kock :dim / 3 :prof - 1
lt 60 kock :dim / 3 :prof - 1
end
```

Kock parallelo

```
turtles-own[generazione lun]
to lato :max
if generazione = :max [ stop]
if generazione = :max - 1 [pd ]
hatch [ set generazione generazione + 1 set lun lun / 3 ]
fd lun lt 60
  hatch [set generazione generazione + 1 set lun lun / 3 ]
fd lun rt 120
hatch [ set generazione generazione + 1 set lun lun / 3 ]
fd lun lt 60
  hatch [set generazione generazione + 1 set lun lun / 3 ]
fd lun
die
end
```

Frattale di Hilbert



Due tipi di tartarughe:

- quelle che formano il ferro di cavallo
- quelle che uniscono i ferri

Codifica Morse

```
__extensions [ "sound.jar" ]  
to .  
  start-note "VIOLIN" 60 64  
  wait 0.1  
  stop-note "VIOLIN" 60  
  wait 0.1  
end  
  
to _  
  start-note "VIOLIN" 60 64  
  wait 0.5  
  stop-note "VIOLIN" 60  
  wait 0.5  
end
```

```
to suona [par]  
  if empty? par [stop]  
  run first par  
  suona bf par  
end
```

Formazione acqua

```
turtles-own[reagente]
breeds [ossigeno idrogeno doppio
       acqua ]
to spargiti
seth random 360
setxy random screen-width random
       screen-height
end

to interact
;vediamo se sullo stesso patch ci sono
       ;molecole;che possono interagire
setreagente one-of-turtles-here
if reagente < 0 [stop]
if color = red [make-h2]
if color = white [make-h2o]
end
```

```
to inizia
ca
create-idrogeno-and-do idro
[setc red spargiti]
create-ossigeno-and-do ossi
[setc yellow spargiti]
end
```


Formazione acqua

to make-h2

;combina due elementi idrogeno

if (color-of reagente) = red

[setc-of reagente white

die]

end

to make-h2o

;combina una molecola H2 con
ossigeno

if (color-of reagente) = yellow

[setc-of reagente green

die]

end

to passeggia

if color = green [stop]

rt random 40

lt random 40

fd 1

interact

passeggia

end

Una nuova primitiva

Far funzionare il programma per formare 50 molecole d'acqua

Contare le molecole: Cosa succede???

to interact

setreagente one-of-turtles-here

grab reagente [if reagente < 0 [stop]

if color = red [make-h2]

if color = white [make-h2o]]

end