Com ::= Block

Block ::= { Declist Comlist }

{

⋮

{ • dich. LOCALI

. . . . .

comandi

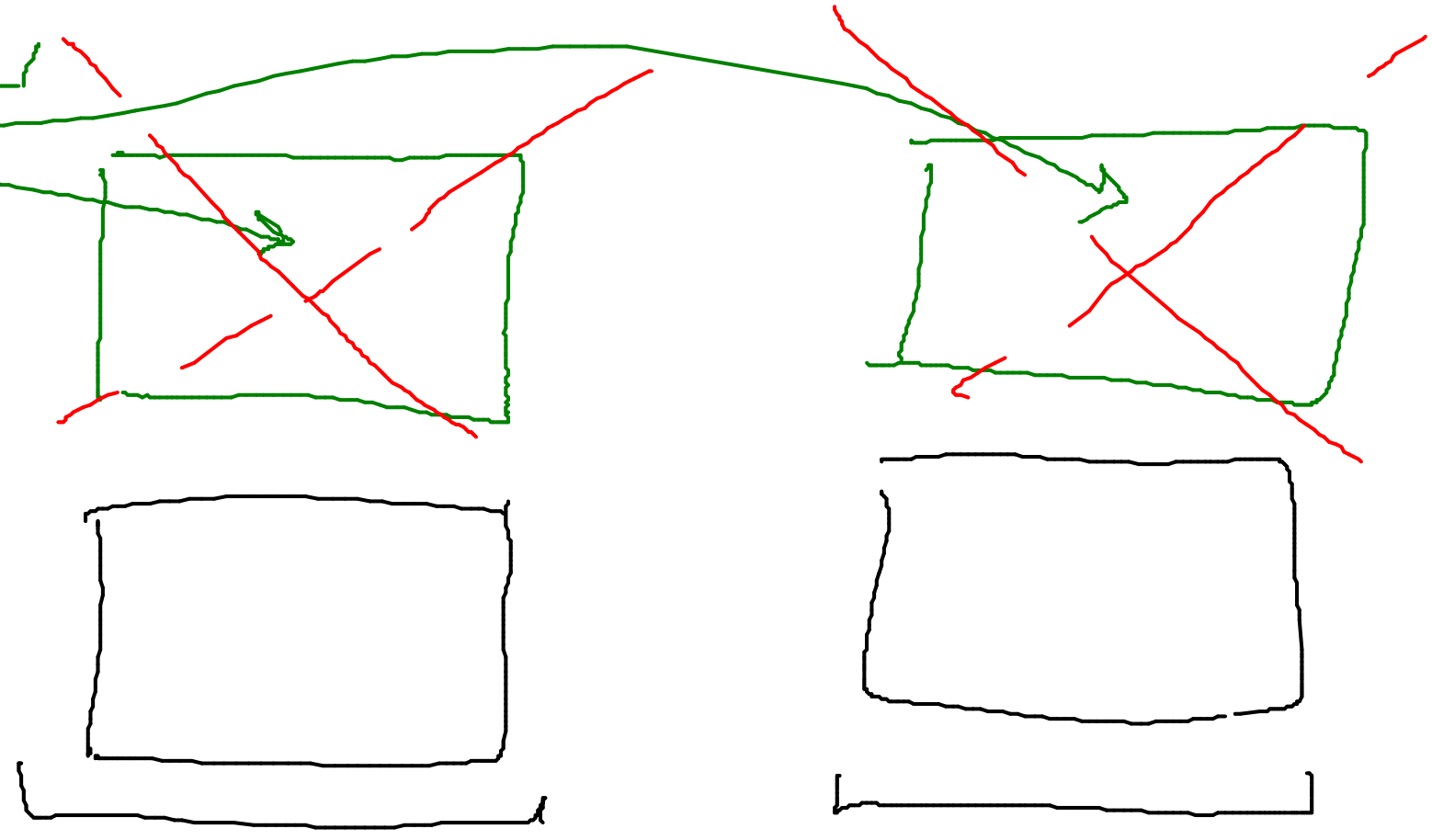
. . . . .

blocco annidato

}

⋮

}



$$Sem_{clist} \quad Comlist \rightarrow P \rightarrow M \rightarrow M$$

$$Sem_{clist}\ c\ \rho\ \mu = Sem_c\ c\ \rho\ \mu \qquad c \in Com$$

$$Sem_{clist}\ [c\ cs]\ \rho\ \mu = Sem_{clist}\ cs\ \rho\ \mu'$$

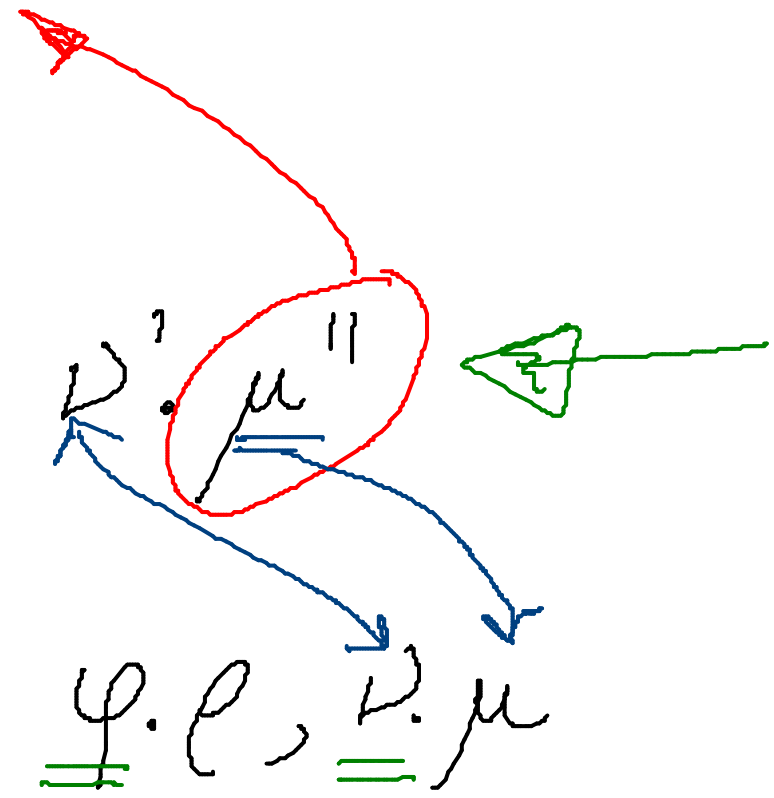
above

$$\mu' = Sem_c\ c\ \rho\ \mu$$

$$Sem_c\ \{ds\ \ cs\}\ \rho\ \mu = \mu''$$

$$Sem_{clist}\ cs\ \ \varphi.\rho\ \nu.\mu = \nu'.\mu''$$

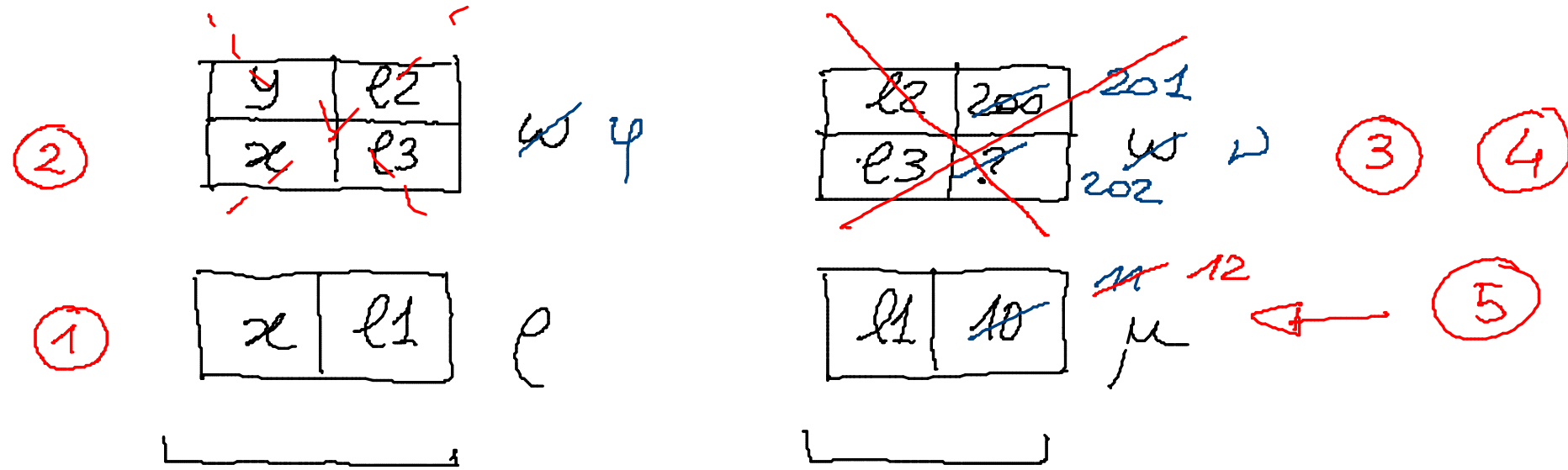
$$Sem_{dlist}\ \underline{ds}\ \ \underline{\omega}.\rho\ \underline{\omega}.\mu = \underline{\varphi}.\rho, \underline{\nu}.\mu$$



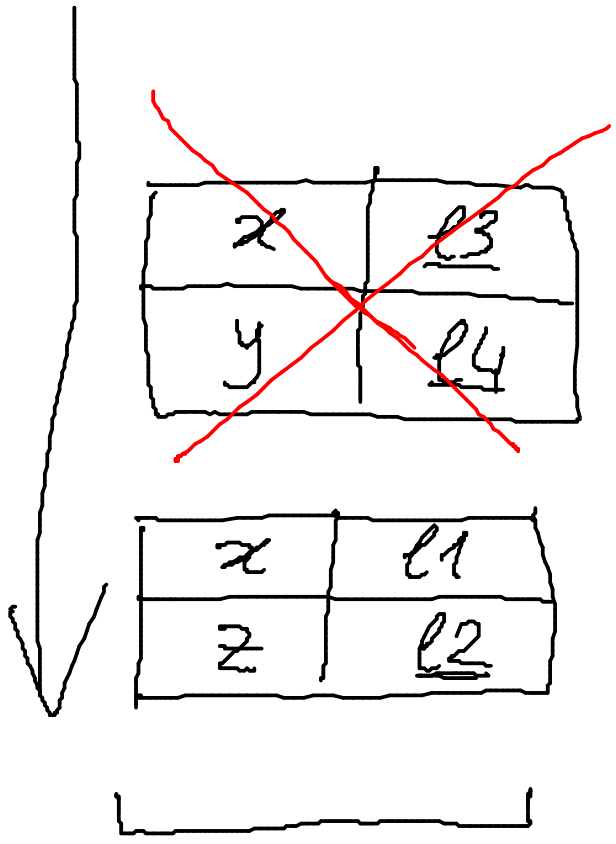
1) aggiunta di un frame vuoto sulle pile $\omega.\rho, \omega.\mu$
2) valutazione delle DICHIARAZIONI LOCALI $\varphi.\rho, \nu.\mu$
3) esecuzione della sequenza di comandi: $\nu'.\mu''$
4) uscita dal blocco: μ''

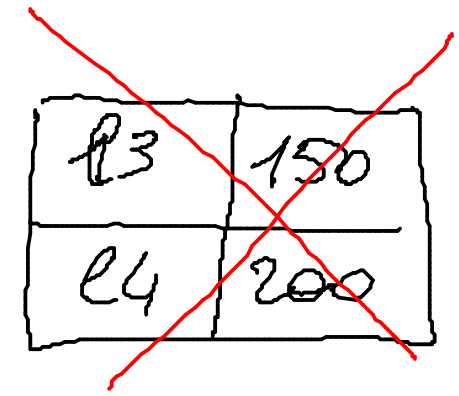
```
{ int x = 10;
  x = x+1;
  { int y = 200;
    int x;
    y = y + 1;
    x = 1 + y;
  }
  x = x + 1;
}
```

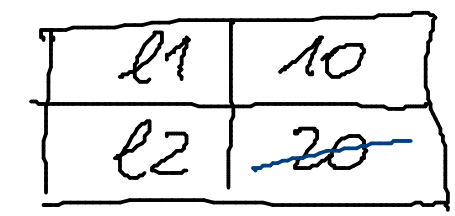
1
2
3
4
5



```
{ int x = 10;
  int z = 20;
    { int x = 150;
      int y = 200;
      z = x + y;
    }
  z = x + z ;
}
```







350
360

$$Sem_{clist}\ \{clist\}\ \rho\ \mu = \mu''$$

$$Sem_{cl}\ Clist\ \ w.\rho\ \ w.\mu = w.\mu''$$

PROGRAMMA

$Prog ::= Block$

$Sem_{prog} : Prog \longrightarrow M$

$Sem_{prog} \{ ds \quad cs \} = Sem_c \{ ds \quad cs \} \Omega \Omega$

attenzione ....

$$Sem_{prog}\{ds\ cs\}$$

$$= Sem_c\{ds\ cs\}\ \Omega\ \Omega$$

$$= \{ \text{def di } Sem_c \quad Sem_{dcst}\ ds\ \omega.\Omega\ \omega.\Omega = \varphi.\Omega, \nu.\Omega \quad Sem_{cdst}\ cs\ \varphi.\Omega\ \nu.\Omega = \nu'.\underline{\underline{\Omega}}$$

$$\Omega$$

la Semantica di un programma è la PILA VUOTA !!

In C e negli altri linguaggi esistono costrutti per COMUNICARE all'esterno i risultati del calcolo (stampa, memorizzazione su periferiche, ecc....)

PUNTATORI in C

Variabili `int x = 10;` caratterizzate da

- un nome
- un valore — componente MODIFICABILE
- un tipo
- un indirizzo — locazione associata nell'ambiente alla variabile

- I puntatori sono VARIABILI come tutte le altre il cui valore è l'INDIRIZZO (locazione) di un'altra variabile

```
{ int  x = 10;
  int *p;
```

← variabile intera

← variabile di tipo PUNTATORE a INTERO:

- durante l'esecuzione potrà assumere come valori indirizzi (locazioni) associate a variabili di tipo int.

```
  p = &x;
  .
  .
  .
 *p = 300;
```

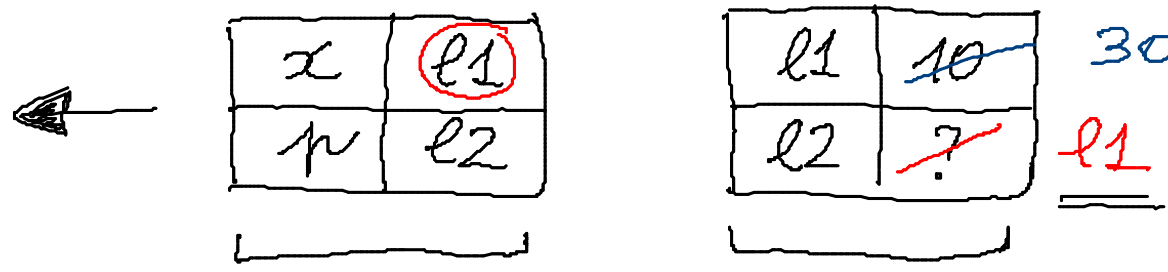
- attraverso un PUNTATORE è possibile modificare il valore della VARIABILE il cui indirizzo è il valore associato al puntatore

denota l'indirizzo di x (la locazione di x)

---

p PUNTA a x

x è PUNTATA da p



```
{ int *p;
  int x = 10;
}
```

```
{ int *p = &x;
  int x = 10;
}
```

x non c'è nello stato

```
{ int x = 10;
  int *p = &x;
}
```

```
{ int x = y+1;
  int y = 20;
}
```

y non c'è nello stato

```
{ int x = 10;
  int *p; int **q;
  p = &x;
  q = &p;
}
```

- &x : l1
- x : 10
- p : l1
- &p : l2

p	l2
x	l1
q	l3

l2	l1
l1	10
l3	l2

ESTENSIONI SINTASSI e SEMANTICA

per TRATTARE PUNTATORI

1) $\mu : Loc \longrightarrow (Val \cup Loc)_{\perp}$

$Val = \mathbb{N} \cup \mathbb{Z} \cup \mathbb{R} \cup \mathbb{C}$

[Type ::= int | int *

[Type ::= BType | PType
BType ::= int | char | float | ....
PType ::= BType *

$Exp ::= \&Ide \mid *Ide$

$Com ::= *Ide = Exp;$

$x = x + 1;$

$*y = *y + 1;$

```
{ int  x = 10;  int z = 30;
  int *p;
```

la variabile incrementata è x

```
x = x + 1;
p = &x;
*p = *p + 1;
```

la variabile modificata è x

```
x = x + 1;
p = &z;
*p = *p + 1;
```

la variabile modificata è z

ESTENSIONI SEMANTICHE

$$Sem_{exp} : Exp \to P \to M \to Val \cup Loc$$

$$Sem_{exp}\ \&x\ \rho\ \mu = \rho\ x \qquad x \in Ide$$

$$Sem_{exp}\ \underline{\underline{*x}}\ \rho\ \mu = \underline{\underline{\mu}}\ (\mu\ (\rho\ x)) \qquad x \in Ide$$

$$Sem_{exp}\ x\ \rho\ \mu = \mu\ (\rho\ x)$$

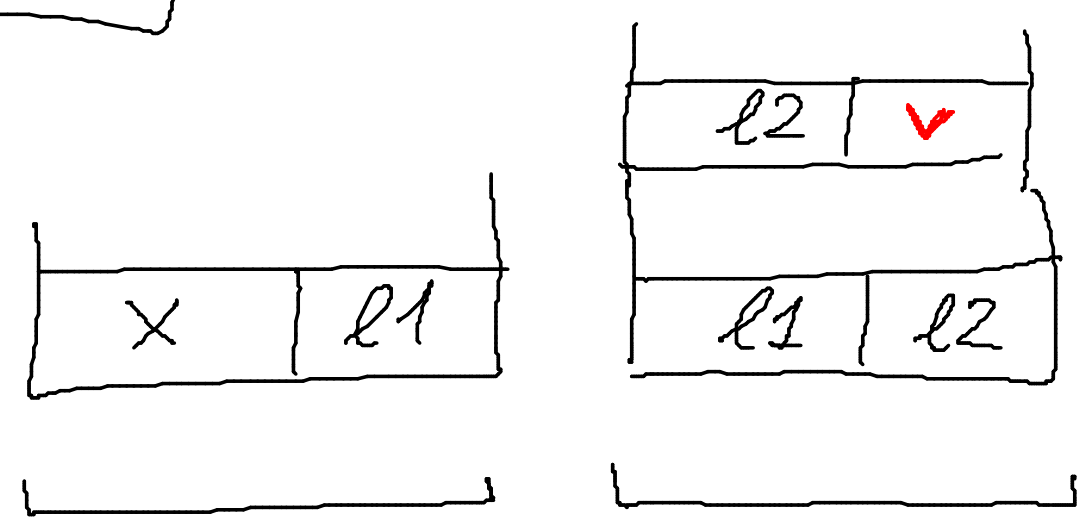
x	$l1$

$l2$	
$l1$	$l2$

$$Sem_{Com}\ *x = e;\ \rho\ \mu = \mu\left[\ \underline{\underline{\frac{v}{\mu(\rho x)}}}\ \right]^{mod}$$

done $v = Sem_{exp}\ e\ \rho\ \mu$

$$Sem_{Com}\ x = e\ \rho\ \mu = \mu\left[\ \frac{v}{\rho x}\ \right]$$



$$\mathrm{Sem}_d\ \ T\ *x;\ \ell\ \mu\ =\ \ell\left[\frac{l}{x}\right]^{\mathrm{add}},\ \mu\left[\frac{?}{l}\right]^{\mathrm{add}}$$

$$l = \mathrm{succloc}\ \mu$$

$$\mathrm{Sem}_d\ \ \underline{\underline{T\ *x = e}}\ \ \ell\ \mu\ =\ \ell\left[\frac{l}{x}\right]^{\mathrm{add}},\ \mu\left[\frac{l'}{l}\right]^{\mathrm{add}}$$

$$\text{done}\quad \mathrm{Sem}_{\mathrm{exp}}\ e\ \ell\ \mu = l'$$