Programmazione I

Lezione 23
22/11/22

Tipi composti

- **puntatori**
- struct
- **array**
- slice
- map
- tipi funzione

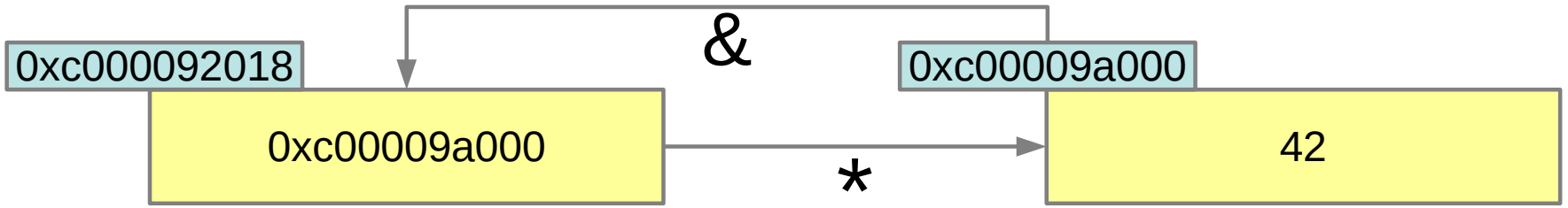
Puntatori

- Recap: tipo definisce valori, operazioni e **rappresentazione** (compresa l'occupazione in memoria)
- Esempio: int32 VS rune VS float64
- Definizioni: locazioni, indirizzi e contenuto
- Tipo puntatore: manipolare direttamente gli indirizzi
- Sintassi GO:

  & → referenziazione (referencing)

  * → indirezione o dereferenziazione (dereferencing)

- *Meccanismo dell'allocazione esplicita: new()*



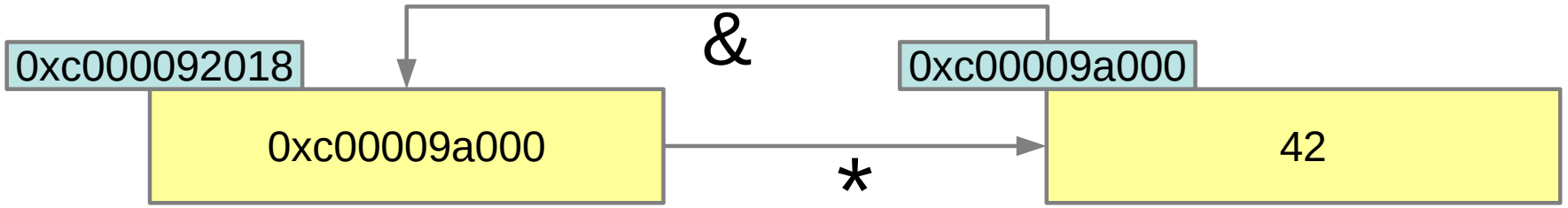
Puntatori

- Sintassi GO:

  var p *int → dichiarazione

  p = &x → referenziazione (referencing)

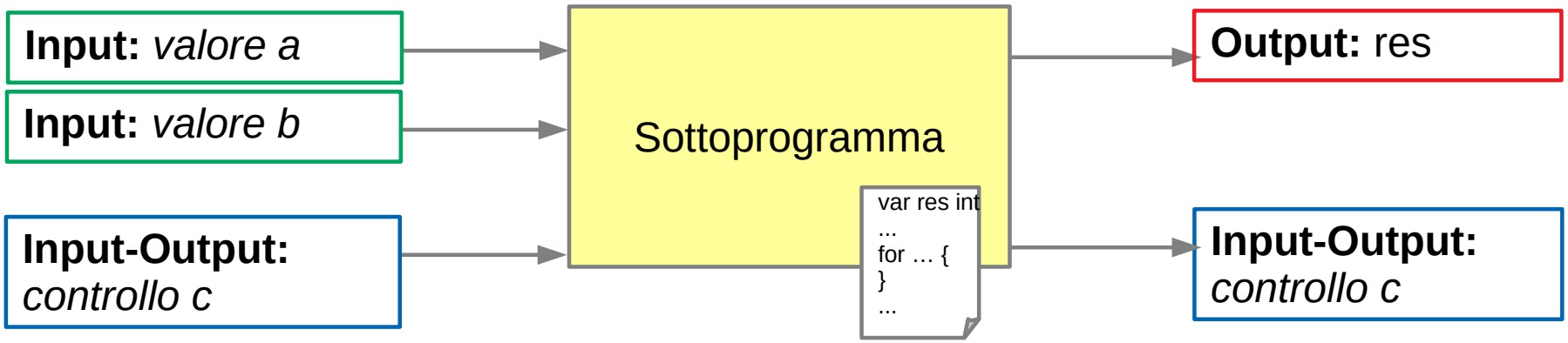
  x = *p → indirezione o dereferenziazione (dereferencing)



- Puntatori per “simulare” passaggio per riferimento (mod. I/O)
- Puntatori ed allocazione esplicita della memoria

Tornando al ruolo dei parametri

- Ruolo dei parametri (agli occhi del programmatore): Input, Output, Input-Output
- Discussione: passaggio per copia, passaggio per reference, uso di puntatori



Puntatori

- Esempio: funzione che incrementa di 2 il valore di una variabile int
- Discussione: passaggio per copia, passaggio per reference, uso di puntatori

Tipi composti

- **puntatori**
- struct
- **array**
- **slice**
- map
- tipi funzione

Array e Slice

- Idea: sequenza di variabili dello stesso tipo, simili ai vettori in matematica

- Array: sequenza la cui lunghezza è fornita in fase di dichiarazione, e non cambia (i.e. dimensione statica)
- Slice: sequenza la cui lunghezza non è fornita necessariamente in fase di dichiarazione, e che può cambiare (i.e. dimensione dinamica)
- Sintassi GO (slice_intro.go)

```
var a, b [10]int

var s []int

...

s = make([]int, N)
```