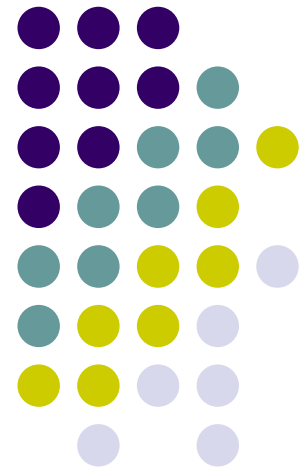


Uso dello stack nella valutazione delle espressioni aritmetiche

Corso: Strutture Dati

Docente: Annalisa De Bonis



Valutare le espressioni aritmetiche



- Quando si valuta un'espressione aritmetica bisogna tener conto dell'ordine di precedenza e dell'associatività degli operatori
 - Qui consideriamo espressioni con operatori associativi a sinistra
- **Idea:** facciamo il push di ogni operatore in uno stack
 - Due stack:
 - `valStack` per i valori
 - `opStack` per gli operatori
 - Si fa il push di ogni operatore ma solo dopo aver estratto e applicato tutti gli operatori con precedenza maggiore o uguale

Algoritmo EvalExp()



Input: una stringa che rappresenta un'espressione aritmetica

Output: il valore dell'espressione

while c'è un altro carattere **c** da scandire

if **c** è un numero **then** valStack.push(**c**)

else

while(opStack non è vuoto \wedge precedenza(**c**) \leq precedenza(opStack.top()))

x \leftarrow valStk.pop(); //fino a che al top dello stack ci sono operatori con

y \leftarrow valStk.pop(); //priorità maggiore rispetto a **c**, si estrae il pop

op \leftarrow opStk.pop(); // di opStack e si applica l'operatore estratto ai

 valStack.push(**y op x**); // due operandi in cima a valStack

opStack.push(**c**)

while(opStack non è vuoto) //alla fine si applicano gli operatori rimasti in opStack

x \leftarrow valStk.pop();

 //agli operandi rimasti in vaStack

y \leftarrow valStk.pop();

op \leftarrow opStk.pop();

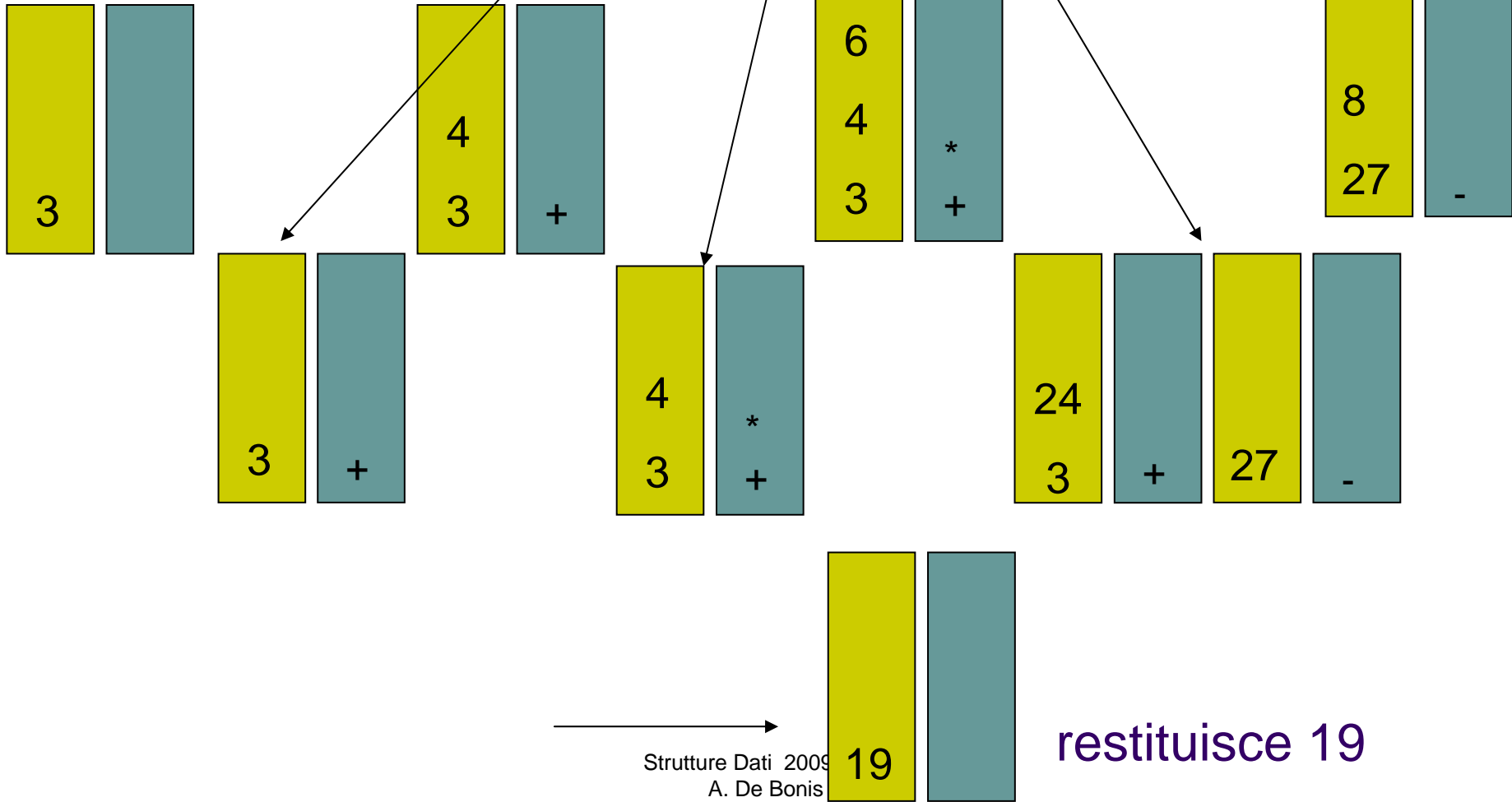
 valStack.push(**y op x**);

return valStack.top();

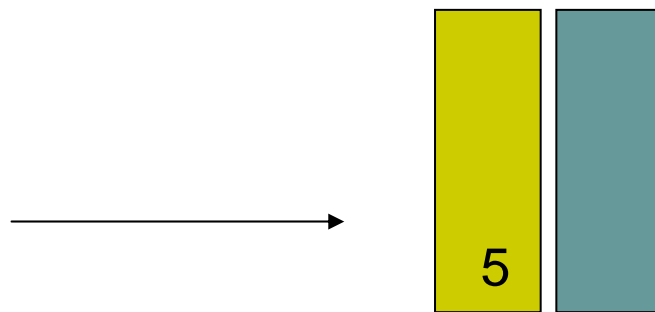
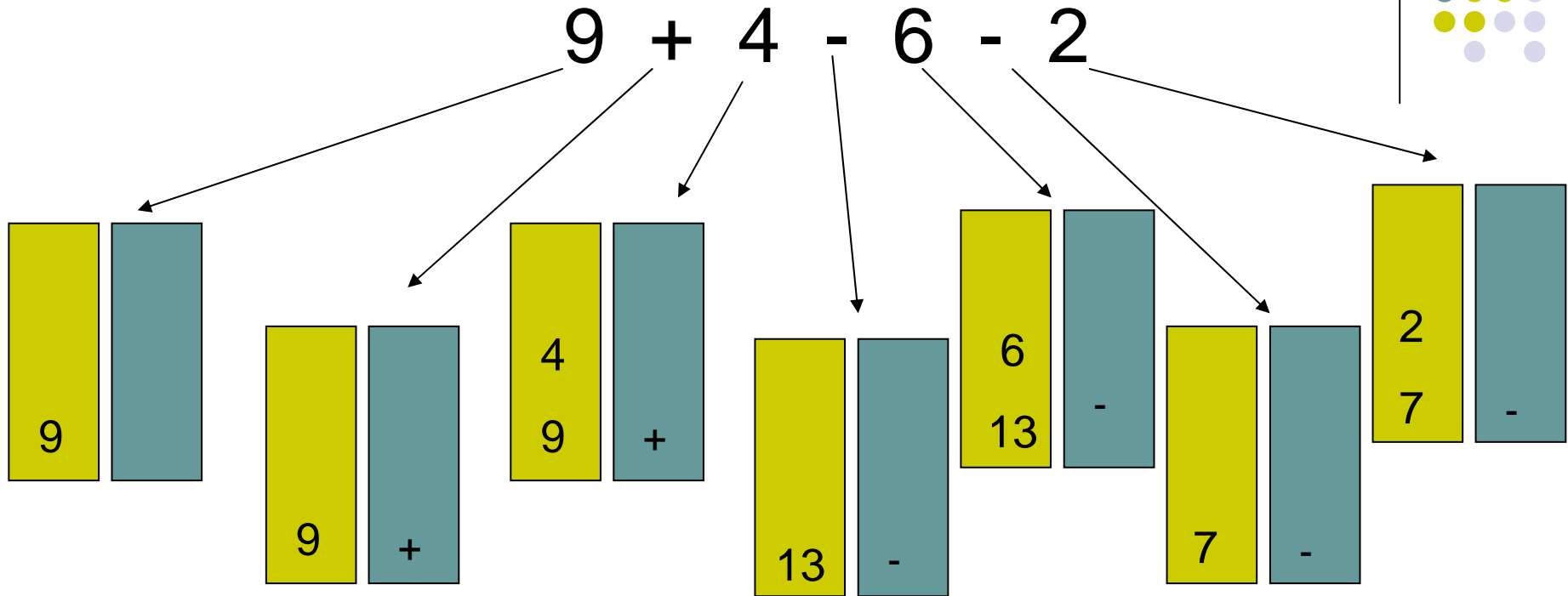
Esempio



3 + 4 * 6 - 8



Esempio



restituisce 5



Esercizio

- Se l'espressione è in notazione postfissa è possibile valutarla usando un unico stack?
- Esempi di espressioni aritmetiche in notazione postfissa
 - $1\ 2\ +\ 5\ *$ corrisponde a $(1 + 2) * 5$
 - $1\ 2\ 5\ * +$ corrisponde a $1 + 2 * 5$