

# [CI2] Concepts informatiques

## 2.1 Piles

Transformations entre les formes infixe, postfixe et préfixe d'une expression arithmétique

---

Daniela Petrişan  
Université de Paris, IRIF



INSTITUT  
DE RECHERCHE  
EN INFORMATIQUE  
FONDAMENTALE



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

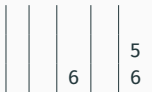
on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$

de l'expression  $6 * (5\%3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$





# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

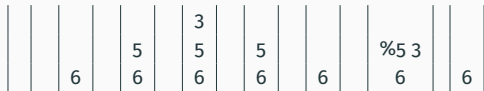
on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4 \ -$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4 \ -$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4 \ -$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

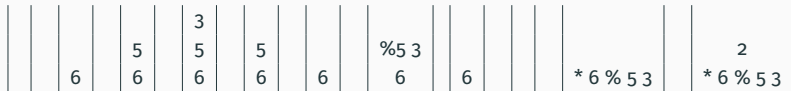
on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4 \ -$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4 \ -$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4 \ -$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

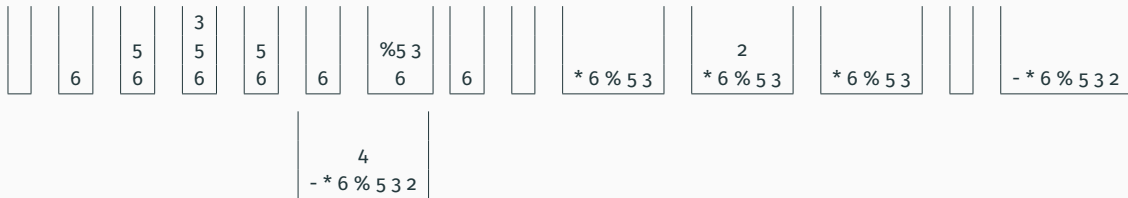
on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $653\% * 2 - 4 -$   
de l'expression  $6 * (5\%3) - 2 - 4$





# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

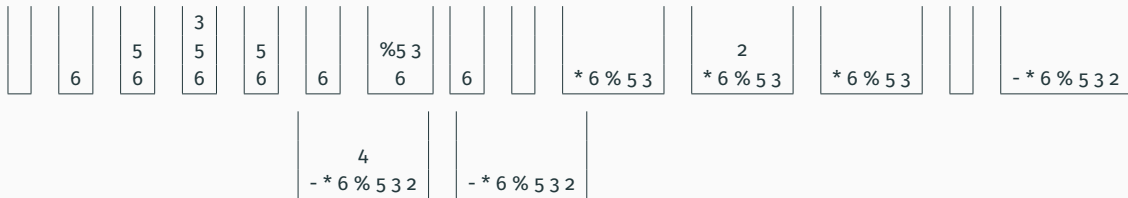
on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4$   
de l'expression  $6 * (5 \% 3) - 2 - 4$



# Algorithme de transformation d'une forme postfixe en forme préfixe

on lit la forme postfixe de gauche à droite  
si un élément est un opérande, on l'empile  
sinon (c'est un opérateur (binaire))

on dépile (deux) éléments

on fait la forme préfixe (opérateur, deuxième élément dépilé, premier élément dépilé)

on empile son résultat

à la fin, la pile contient la forme préfixe

forme postfixe :  $6\ 5\ 3\ \% \ * \ 2 \ - \ 4$   
de l'expression  $6 * (5 \% 3) - 2 - 4$

