

Notes de cours

GIF 340 : ÉLÉMENTS DE COMPILATION

Chapitre 7

*Évaluation des représentations :
infix, postfix, AST*

Ahmed KHOUMSI

Plan

- Représentation infix
- Opérandes, précédence et associativité d'opérateurs
- Représentation postfix et AST
- Évaluation d'AST
- Évaluation d'expression postfix
- Conversion de infix à postfix

Représentation infix

Expression infix constituée de :

- opérandes
- opérateurs (binaires ou unaires)
- parenthèses

Les deux opérandes d'un opérateur binaire sont respectivement à droite et à gauche de l'opérateur

Exemples : $(5 + 7)$ $3/2$ 2^3

L'opérande d'un opérateur unaire est à droite de l'opérateur

Exemple : -7

Opérandes et opérateurs considérés

Opérandes : nombres entiers

Opérateurs binaires

- | | |
|----------|----------------------------|
| \wedge | exposant |
| $*$ $/$ | multiplication et division |
| $+$ $-$ | addition et soustraction |

Opérateur unaire : $-$

Exemple : $1 - 3 - 2 * 7 \wedge 4 * 5 / 6 \wedge 2 \wedge 3$

Précédence et associativité des opérateurs

Précédence (ou priorité)

Soient deux opérateurs op1 et op2

Si op1 est plus prioritaire que (*a un précédence sur*) op2, alors :

« a op1 b op2 c » doit être interprété comme « (a op1 b) op2 c »

Associativité : à droite, à gauche

Soient deux opérateurs op1 et op2 de même priorité et de même associativité

Si op1 et op2 sont associatifs à gauche, alors :

« a op1 b op2 c » doit être interprété comme « (a op1 b) op2 c »

Si op1 et op2 sont associatifs à droite, alors :

« a op1 b op2 c » doit être interprété comme « a op1 (b op2 c) »

Précédence et associativité (suite)

Opérateurs par ordre de précédence

Opérateur	Type	Associativité
^	binaire	à droite
-	unaire	à droite
* /	binaire	à gauche
+ -	binaire	à gauche

Exemple : l'expression infix

$1 - 2 - 3 * 4 ^ 5 * 6 / 7 ^ 2 ^ 2$

est interprétée comme l'expression infix suivante :

$(1 - 2) - (((3 * (4 ^ 5)) * 6) / (7 ^ (2 ^ 2)))$

Remarque : le moins unaire n'est pas considéré dans la suite de ce chapitre, mais vous devrez le traiter dans le second laboratoire

Représentation postfix

Séquence d'opérateurs et d'opérandes

Les positions des opérateurs et opérandes sont telles que l'expression puisse être évaluée :

- sans mettre de parenthèse
- sans connaître les priorités et associativités des opérateurs

Les positions des opérateurs et opérandes informent implicitement sur les priorités et associativités

Position d'un opérateur est toujours plus à droite des positions de ses opérandes

Exemple : Soit l'expression infix $1 + 2 * 3$

Si $*$ est plus prioritaire que $+$ alors l'expression postfix correspondante est $1\ 2\ 3\ *\ +$

Si $*$ est moins prioritaire que $+$ alors l'expression postfix correspondante est $1\ 2\ +\ 3\ *$

Passage de infix à : - arbres syntaxiques abstraits (AST) - expression postfix

Passage de infix à AST

Soit l'expression infix $1 + 2 * 3$ dont le résultat est :

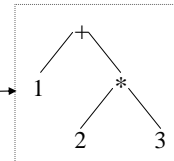
7 si $*$ est plus prioritaire que $+$

9 si $*$ est moins prioritaire que $+$

$*$ plus prioritaire que $+$

L'expression infix peut être représentée par l'AST

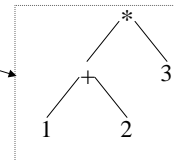
L'expression postfix correspondante est $1\ 2\ 3\ *\ +$



$*$ moins prioritaire que $+$

L'expression infix peut être représentée par l'AST

L'expression postfix correspondante est $1\ 2\ +\ 3\ *$



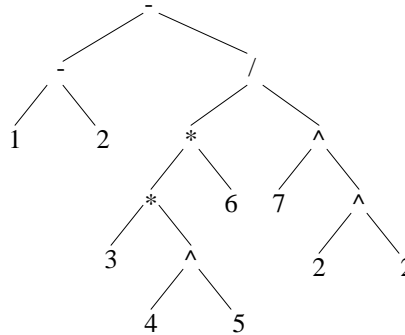
Passage de infix à AST (suite)

Soit l'expression infix $1 - 2 - 3 * 4 ^ 5 * 6 / 7 ^ 2 ^ 2$

En appliquant les règles de priorité et d'associativité (voir pages 5 et 6),
cette expression est interprétée comme :

$$(1 - 2) - (((3 * (4 ^ 5)) * 6) / (7 ^ (2 ^ 2)))$$

L'AST correspondant est :

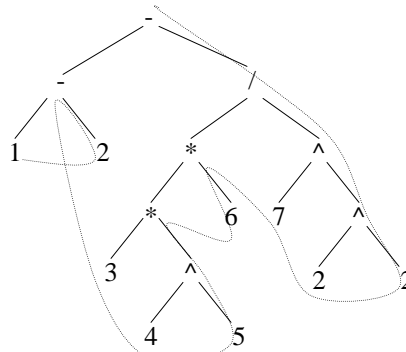


Passage de AST à expression postfix (suite)

- parcourir feuilles de gauche à droite
- remonter à un nœud à partir de son enfant le plus à droite
(càd lorsque tous les enfants du nœud ont été parcourus)

Exemple : L'expression postfix correspondant à l'AST ci-dessous est :

$$1 2 - 3 4 5 ^ * 6 * 7 2 2 ^ ^ / -$$



Évaluation d'AST

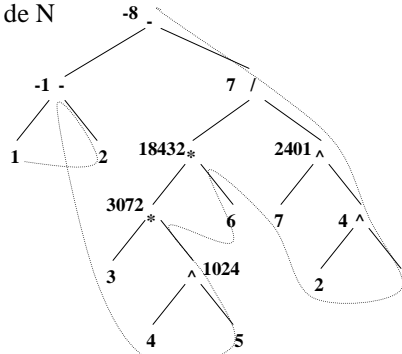
Pour simplifier, on considère le cas où un AST est constitué de feuilles et de nœuds tels que :

- une feuille a une valeur
- un nœud est défini par un opérateur binaire et ses enfants gauche et droit
- Évaluation d'une feuille = valeur de la feuille
- Évaluation d'un nœud = valG Op valD, où :
 - Op est l'opérateur du nœud
 - valG est l'évaluation de l'enfant gauche de N
 - valD est l'évaluation de l'enfant droit de N

Nous voyons donc que l'évaluation d'un AST est l'évaluation de sa racine et qu'elle peut se faire d'une manière récursive.

Évaluation d'AST par un parcours postfix

- parcourir les feuilles de gauche à droite
- remonter à un nœud à partir de son enfant le plus à droite (càd lorsque tous les enfants du nœud ont été parcourus)
- s'arrêter lorsqu'on arrive au nœud racine.
- Lorsqu'on arrive à une feuille, on l'évalue
- Lorsqu'on arrive à un nœud N, on l'évalue en appliquant l'opérateur de N aux évaluations des enfants de N



Évaluation d'une expression postfix

Lecture de l'expression postfix de gauche à droite

Lorsqu'un opérande est rencontré, alors il est mis (push) dans une pile

Lorsqu'un opérateur est rencontré, alors :

- le nombre approprié d'opérandes (2, si opérateur binaire) sont récupérés (topAndPop) de la pile
- l'opération est effectuée
- le résultat est mis (push) dans la pile

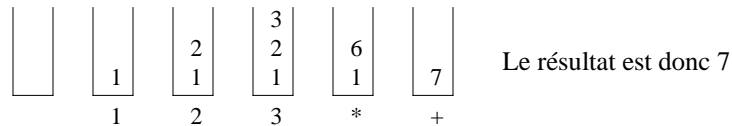
À la fin, la pile devrait contenir un seul élément qui est l'évaluation de l'expression postfix

Il n'est pas nécessaire d'utiliser les règles de priorité et d'associativité, car celles-ci sont implicitement déterminées à partir des positions des opérateurs et des opérandes dans l'expression postfix

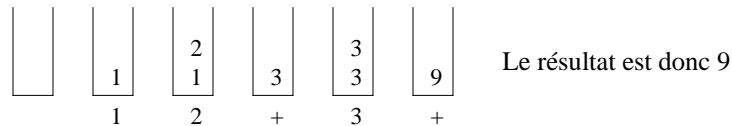
Évaluation d'une expression postfix : exemple 1

Soit l'expression infix $1 + 2 * 3$

Si $*$ est plus prioritaire que $+$, alors l'expression postfix correspondante est :
 $1\ 2\ 3\ *\ +$ (voir page 8)



Si $*$ est moins prioritaire que $+$, alors l'expression postfix correspondante est :
 $1\ 2\ +\ 3\ *$ (voir page 8)

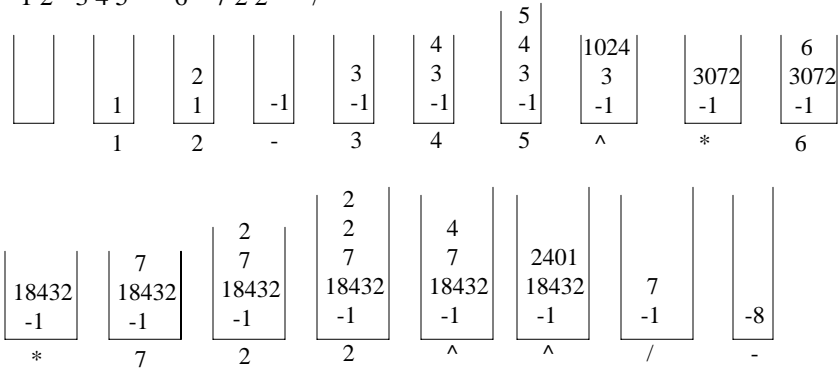


Évaluation d'une expression postfix : exemple 2

L'expression infix $1 - 2 - 3 * 4 ^ 5 * 6 / 7 ^ 2 ^ 2$ qui doit être interprétée comme :
 $(1 - 2) - (((3 * (4 ^ 5)) * 6) / (7 ^ (2 ^ 2)))$

L'expression postfix correspondante est : (voir page 10)

1 2 - 3 4 5 ^ * 6 * 7 2 2 ^ ^ / -



Conversion de infix à postfix

Nous avons déjà montré d'une manière intuitive, comment la conversion est effectuée à l'aide d'un AST

Nous montrons ici comment la conversion est faite d'une manière systématique à l'aide d'une pile

Lecture de l'expression infix de gauche à droite

Opérandes : l'ordre dans lequel ils apparaissent est le même dans les deux expressions

Opérateurs :

- L'ordre dans lequel ils apparaissent peut être différent dans les deux expressions
Cet ordre dépend des priorités et des associativités
- Ils apparaissent après leurs opérandes

Lorsque opérateur rencontré (dans l'expression infix), il est mis dans une pile (opération push)

Il sera récupéré ultérieurement de la pile (opération topAndPop) pour la construction de l'expression postfix

Conversion de infix à postfix : cas général

Lorsqu'un opérande est rencontré, alors il est retourné

Lorsqu'un opérateur est rencontré, alors :

- Tous les opérateurs dans la pile qui ont :
 - une plus grande priorité, ou
 - une même priorité et sont associatifs à gauchesont récupérés de la pile (topAndPop) et retournés
- L'opérateur rencontré est ensuite mis dans la pile (push)

Lorsque toute l'expression infix a été lue, alors le contenu de la pile est récupéré (topAndPop) et retourné

La séquence des symboles retournés constitue l'expression postfix

Conversion de infix à postfix : exemple simple

Soit l'expression infix « $3^2 + 6$ »

Après lecture de « 3^2 » : - l'expression postfix courante est : « 3^2 »
- la pile contient : ^

Après lecture de + :

comme + est moins prioritaire que ^, donc ^ est récupéré de la pile (topAndPop) et mis à droite de l'expression postfix courante.

On obtient « $3^2^$ »

+ est ensuite mis dans la pile (push)

Après lecture de 6

L'expression postfix courante devient : « 3^2^6 »

Comme toute l'expression infix a été lue, on récupère le contenu de la pile (topAndPop) pour le mettre à droite de l'expression postfix courante.

Le résultat final est donc : « $3^2^6^$ »

Conversion de infix à postfix : autre exemple simple

Soit l'expression infix « $1 * 2 ^ 3 + 6$ »

Après lecture de « $1 * 2$ » : - l'expression postfix courante est : « $1 2$ »

- la pile contient : *

Après lecture de $^$: comme $^$ est plus prioritaire que * alors

$^$ est mis dans la pile (push)

Après lecture de 3 : - l'expression postfix courante est : « $1 2 3$ »

- la pile contient : * $^$

Après lecture de $+$:

comme $+$ est moins prioritaire que $^$ et *, donc $^$ et * sont récupérés

de la pile (topAndPop) et mis à droite de l'expression postfix courante.

On obtient « $1 2 3 ^ *$ »

$+$ est ensuite mis dans la pile (push)

Après lecture de 6

L'expression postfix courante devient : « $1 2 3 ^ * 6$ »

Comme toute l'expression infix a été lue, on récupère le contenu de la pile (topAndPop) pour le mettre à droite de l'expression postfix courante.

Le résultat final est donc : « $1 2 3 ^ * 6 +$ »

Cas des parenthèses

Parenthèse ouvrante « (» doit être considérée comme :

- opérateur de plus haute priorité lorsque elle est rencontrée dans expression infix
- opérateur de plus basse priorité lorsqu'elle se trouve dans la pile

Parenthèse fermante «) »

Lorsque rencontrée dans expression infix, alors on récupère (topAndPop)

tous les opérateurs de la pile jusqu'à ce qu'on rencontre une parenthèse ouvrante

- les opérateurs récupérés sont retournés (pour être mis dans l'expression postfix)
- les deux parenthèses ne sont pas retournées

Algorithme de conversion : récapitulation

Opérandes : immédiatement retournés lorsque rencontrés

Parenthèse fermante : lorsque rencontrée :

- il faut récupérer et retourner tous les symboles de la pile jusqu'à ce qu'on trouve une parenthèse ouvrante
- la parenthèse ouvrante est récupérée de la pile mais n'est pas retournée
- la parenthèse fermante n'est pas retournée

Opérateur (incluant parenthèses ouvrantes) lorsque rencontré :

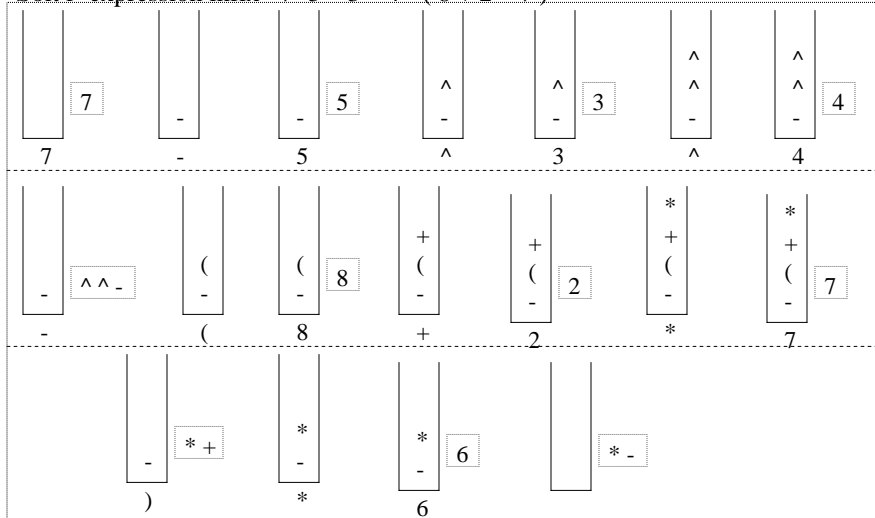
- Récupérer et retourner tous les symboles de la pile jusqu 'à ce qu 'on trouve un opérateur :
 - de plus haute priorité, ou
 - de même priorité et associatif à droite
- Celui-ci 'est ni récupéré ni retourné
- Ensuite, mettre dans la pile l 'opérateur rencontré

Fin de lecture : lorsque expression infix entièrement lue, récupérer et retourner tout le contenu de la pile

Rappel : parenthèses ouvrantes traitées comme opérateurs particuliers
(voir page précédente)

Algorithme de conversion de infix à postfix : exemple

Soit l'expression infix $7-5 \wedge 3 \wedge 4-(8+2 * 7)$



L'expression postfix est donc : 7 5 3 4 ^ ^ - 8 2 7 * + 6 * -