

# Rappels

## Expressions Arithmétiques Générales (EAG) (TPs 3 et 4)

- opérateurs  $+$ ,  $-$ ,  $\times$  avec priorités
- parenthésage possible

Exemples :  $42$ ,  $2 * 5 + 3$ ,  $2 * (5 + 3)$ , etc.

## Les étapes

- 1 Analyse Lexicale : pas de changement par rapport au TP2 (ENTIER, PLUS, MOINS, MUL, DIV, PARO, PARF)
- 2 Analyse Syntaxique :
  - 1 écrire la **grammaire** du langage  
→ vu en cours la semaine dernière ...
  - 2 écrire l'analyseur syntaxique à partir de cette grammaire
- 3 Evaluer l'expression :
  - 1 construire l'**arbre abstrait** lors de l'analyse syntaxique
  - 2 parcours de l'Ast ppur calculer le résultat

# Rappels

## Expressions Arithmétiques Générales (EAG) (TPs 3 et 4)

- opérateurs  $+$ ,  $-$ ,  $\times$  avec priorités
- parenthésage possible

Exemples :  $42$ ,  $2 * 5 + 3$ ,  $2 * (5 + 3)$ , etc.

## Les étapes

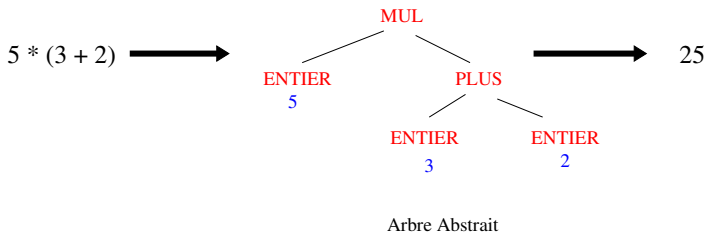
- ➊ Analyse Lexicale : pas de changement par rapport au TP2 (ENTIER, PLUS, MOINS, MUL, DIV, PARO, PARF)
- ➋ Analyse Syntaxique :
  - ➊ écrire la **grammaire** du langage  
→ vu en cours la semaine dernière ...
  - ➋ écrire l'analyseur syntaxique à partir de cette grammaire
- ➌ Evaluer l'expression :
  - ➊ construire l'**arbre abstrait** lors de l'analyse syntaxique
  - ➋ parcours de l'Ast ppur calculer le résultat

# Grammaire des EAG

$eag \rightarrow seq\_terme$   
 $seq\_terme \rightarrow terme \ suite\_seq\_terme$   
 $suite\_seq\_terme \rightarrow op1 \ terme \ suite\_seq\_terme$   
 $suite\_seq\_terme \rightarrow \varepsilon$   
 $terme \rightarrow seq\_facteur$   
 $seq\_facteur \rightarrow facteur \ suite\_seq\_facteur$   
 $suite\_seq\_facteur \rightarrow op2 \ facteur \ suite\_seq\_facteur$   
 $suite\_seq\_facteur \rightarrow \varepsilon$   
 $facteur \rightarrow ENTIER$   
 $facteur \rightarrow PARO \ eag \ PARF$   
 $op1 \rightarrow PLUS$   
 $op1 \rightarrow MOINS$   
 $op2 \rightarrow MUL$

## Reste donc à voir ...

- 1 Ecrire l'analyseur syntaxique pour le langage *eag* à partir de la Grammaire des EAG.
- 2 Etendre cet analyseur pour construire **l'arbre abstrait** des expressions lues.
- 3 Parcourir de cet arbre abstrait pour calculer la valeur de l'expression.



# Ecrire l'analyseur à partir de la Grammaire?

On “suit” les règles de la grammaire (c.g. TP2 sur les EAEP)

*eag* → *seq\_terme*  
*seq\_terme* → *terme suite\_seq\_terme*  
*suite\_seq\_terme* → *op1 terme suite\_seq\_terme*  
*suite\_seq\_terme* →  $\varepsilon$   
*op1* → *PLUS*  
*op1* → *MOINS*

```
Rec_eag ()  
  Rec_seq_terme()
```

```
Rec_seq_terme()  
  Rec_terme() ; Rec_suite_seq_terme()
```

```
Rec_suite_seq_terme ()  
  si LC.nature = PLUS ou LC.nature = MOINS alors  
    Rec_op1() ; Rec_term() ; Rec_suite_seq_terme()
```

# Ecrire l'analyseur à partir de la Grammaire?

On “suit” les règles de la grammaire (c.g. TP2 sur les EAEP)

$$\begin{aligned} eag &\rightarrow seq\_terme \\ seq\_terme &\rightarrow terme \text{ suite\_seq\_terme} \\ suite\_seq\_terme &\rightarrow op1 \text{ terme suite\_seq\_terme} \\ suite\_seq\_terme &\rightarrow \varepsilon \\ op1 &\rightarrow PLUS \\ op1 &\rightarrow MOINS \end{aligned}$$

```
Rec_eag ()  
    Rec_seq_terme()
```

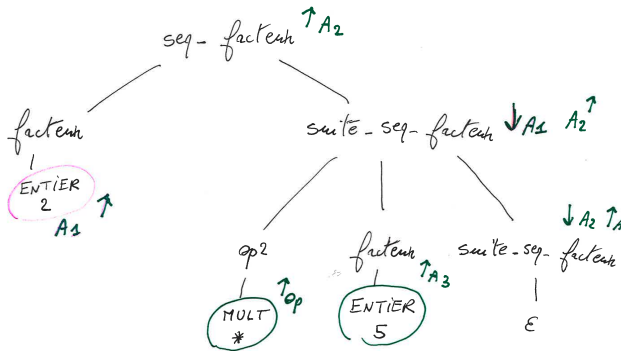
```
Rec_seq_terme()  
    Rec_terme() ; Rec_suite_seq_terme()
```

```
Rec_suite_seq_terme ()  
    si LC.nature = PLUS ou LC.nature = MOINS alors  
        Rec_op1() ; Rec_term() ; Rec_suite_seq_terme()
```

# Principe de construction de l'AST

- synthétiser des “fragments” de l'AST lors de l'analyse de certaines règles
- les propager pour les combiner lors de l'analyse de certaines (autres !) règles

Exemple : construire l'AST de  $2 * 5$  à partir de *seq\_facteur*



$$A_2 = \text{check-operation}(A_1, \text{op}, A_3)$$

# Spécifier la construction sur la Grammaire ?

On “décore” les règles avec des **attributs** *herités* et *synthétisés*

$$\text{facteur}^{\uparrow A1} \rightarrow \text{ENTIER}$$
$$A1 = \text{creer\_valeur}(\text{LC.valeur})$$

L'AST  $A1$  renvoyé par *facteur* est celui produit à partir du lexème courant

$$\text{seq\_facteur}^{\uparrow A2} \rightarrow \text{facteur}^{\uparrow A1} \text{ suite\_seq\_facteur}^{\downarrow A1 \uparrow A2}$$

L'AST  $A2$  renvoyé par *seq\_facteur* est celui produit par *suite\_seq\_facteur* à partir de l'AST  $A1$  produit par *facteur*

$$\text{suite\_seq\_facteur}^{\downarrow A1 \uparrow A2} \rightarrow \text{op2}^{\uparrow Op} \text{ facteur}^{\uparrow A3} \text{ suite\_seq\_facteur}^{\downarrow A4 \uparrow A2}$$
$$A4 = \text{creer\_operation}(Op, A1, A3)$$

L'AST  $A2$  renvoyé par *suite\_seq\_facteur* est celui produit par *suite\_seq\_facteur* à partir de l'AST  $A4$  construit par *creer\_operation*( $Op, A1, A3$ )



# Ecrire l'analyseur : exemple 1

Les attributs deviennent des paramètres des fonctions de l'analyseur ...

$$\textit{facteur}^{\uparrow A1} \rightarrow \textit{ENTIER}$$
$$A1 = \textit{creer\_valeur}(\textit{LC.valeur})$$

```
Rec_facteur (Ast *A1) {  
  switch (lexeme_courant().nature {  
    ...  
    case ENTIER: *A1 = creer_valeur(lexeme_courant().valeur);  
                  avancer() ; break ;  
    ...  
  }  
}
```

## Ecrire l'analyseur : exemple 2

$suite\_seq\_facteur \downarrow A1 \uparrow A2 \rightarrow op2 \uparrow Op \text{ facteur} \uparrow A3 \text{ suite\_seq\_facteur} \downarrow A4 \uparrow A2$   
 $A4 = \text{creer\_operation}(Op, A1, A3)$

```
Rec_suite_seq_facteur (Ast A1, Ast *A2) {  
    Ast A3, A4 ;  
    TypeOperateur Op;  
    if (LC().nature == MUL) {  
        Rec_op2 (&Op) ;  
        Rec_facteur (&A3) ;  
        A4 = creer_operation(Op, A1, A3) ;  
        Rec_suite_seq_facteur (A4, A2) ;  
    } else  
        *A2 = A1 ; // regle epsilon  
}
```

# EAG avec construction de l'AST (1/3)

$$eag \uparrow^{A1} \rightarrow seq\_terme \uparrow^{A1}$$

$$seq\_terme \uparrow^{A2} \rightarrow terme \uparrow^{A1} suite\_seq\_terme \downarrow^{A1} \uparrow^{A2}$$

$$suite\_seq\_terme \downarrow^{A1} \uparrow^{A2} \rightarrow op1 \uparrow^{Op} terme \uparrow^{A3} suite\_seq\_terme \downarrow^{A4} \uparrow^{A2}$$

$A4 = \text{creer\_operation}(Op, A1, A3)$

$$suite\_seq\_terme \downarrow^{A1} \uparrow^{A2} \rightarrow \varepsilon$$

$A2 = A1$

## EAG avec construction de l'AST (2/3)

$$\text{terme}^{\uparrow A1} \rightarrow \text{seq\_facteur}^{\uparrow A1}$$

$$\text{seq\_facteur}^{\uparrow A2} \rightarrow \text{facteur}^{\uparrow A1} \text{ suite\_seq\_facteur}^{\downarrow A1 \uparrow A2}$$

$$\text{suite\_seq\_facteur}^{\downarrow A1 \uparrow A2} \rightarrow \text{op2}^{\uparrow Op} \text{ facteur}^{\uparrow A3} \text{ suite\_seq\_facteur}^{\downarrow A4 \uparrow A2}$$

$A4 = \text{creer\_operation}(Op, A1, A3)$

$$\text{suite\_seq\_facteur}^{\downarrow A1 \uparrow A2} \rightarrow \varepsilon$$

$A2 = A1$

## EAG avec construction de l'AST (3/3)

$\text{facteur}^{\uparrow A1} \rightarrow \text{ENTIER}$

$A1 = \text{creer\_valeur}(\text{LC.valeur})$

$\text{facteur}^{\uparrow A1} \rightarrow \text{PARO } \text{eag}^{\uparrow A1} \text{ PARF}$

$\text{op1}^{\uparrow N\_PLUS} \rightarrow \text{PLUS}$

$\text{op1}^{\uparrow N\_MOINS} \rightarrow \text{MOINS}$

$\text{op2}^{\uparrow N\_MUL} \rightarrow \text{MUL}$