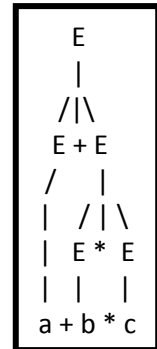
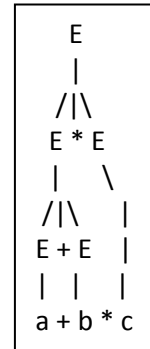


Correction :

A. Grammaires hors contexte

1) La grammaire est bien ambiguë, à partir de E comme symbole non terminal de départ, on peut dériver deux arbres d'analyse différents.

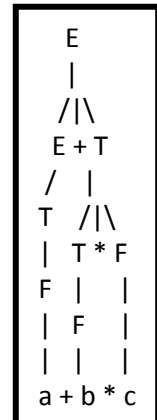


2) Voici une grammaire non ambiguë et qui par ailleurs respecte l'ordre de traitement des opérateurs en arithmétique.

$$E \rightarrow E + T \mid T$$

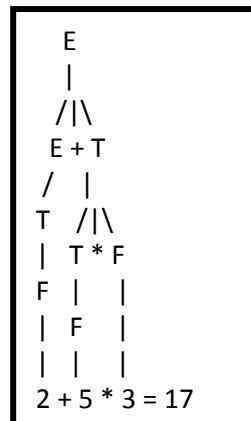
$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid a \mid b \mid c$$



3) Pour tenir compte de l'ordre des opérateurs en arithmétique, il n'y a que l'arbre de dérivation en encadré épais qui est valable. On dérive d'abord les additions/soustraction, puis ensuite les facteurs. (Plus on avance dans l'arbre d'analyse plus on est en mesure de réaliser les calculs intermédiaires et donc les plus prioritaire arrivent au plus près des feuilles).

Autrement présenté, on peut s'intéresser à la sémantique pour comprendre les motivations d'organisation syntaxique de notre grammaire. Concrétisons l'exemple avec des valeurs numériques : 2, 5, 3 (resp. a, b, c). De notre connaissances des règles de l'arithmétique (presque réflexe), nous savons que (sauf à le préciser par des parenthèses) l'ordre dans lequel doivent être traitées les opérations pose le produit comme prioritaire sur l'addition. Notre grammaire « force » l'application de ses règles. Ainsi :



L'autre analyse possible n'est pas consistante avec les règles usuelles de l'arithmétique et reviendrait à calculer : $(2+5) * 3 = 21$

4) On ajoute la soustraction et la division (d'un point de vue seulement syntaxique).

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid a \mid b \mid c$

B. Spécification de formats de données en ABNF

1) CFG des expressions arithmétiques élémentaires de la partie a) en ABNF

// en ABNF

```
expr-arith    =   terme * ( WSP ("+" / "-") WSP terme)
terme         =   facteur * ( WSP ("*" / "/" ) WSP facteur)
facteur       =   DIGIT / WSP "(" expr-arith ")" WSP
```

2) Grammaire ABNF fiche recettes de cuisine

(le lien vers la spécification du langage ABNF : <https://tools.ietf.org/html/rfc5234>)

Grammaire à revoir avec highlight (#1#) :

```
Recette       =   Titre CRLF Auteur CRLF 1## Préparation
Titre         =   TEXT ', ' WSP 'pour' WSP 1*DIGIT 'personnes'
Auteur        =   'par' WSP 2#1*ALPHA#
Ingrédients   =   'Ingrédients :' CRLF 1*Item 3#CRLF#
Item          =   TEXT WSP ':' WSP Quantité 4#Unité#
Quantité      =   1*DIGIT 5##
Unité         =   'c.s.' / 'c.c.' / 'l' / 'g'
Préparation   =   'Préparation 6#:#' CRLF 1*Etape
Etape         =   '*' TEXT CRLF
TEXT          =   1*(WSP / VCHAR)
```

1# Il manque les ingrédients dans la règle Recette

2# Le nom de l'auteur ne doit comprendre que des ALPHA alors que dans l'exemple il comprend également des chiffres (Cuistot12)

3# Le retour à la ligne est mal placé. La ligne doit être sautée entre les ingrédients et pas après la concaténation de tous les ingrédients)

4# Les unités sont optionnelles (ie, 3 oeufs)

5# Quantité ne permet pas d'exprimer des fractions usuelle en cuisine 1/2 3/4 ...

6# Les ':' sont de trop

→ la version corrigée de la grammaire

```
Recette       =   Titre CRLF Auteur CRLF Ingrédients Préparation
Titre         =   TEXT ', ' WSP 'pour' WSP 1*DIGIT 'personnes'
Auteur        =   'par' WSP NAME
Ingrédients   =   'Ingrédients :' CRLF 1*(Item CRLF)
Item          =   TEXT WSP ':' WSP Quantité *1(Unité)
; Item       =   TEXT WSP ':' WSP Quantité [Unité] fonctionne aussi bien
Quantité      =   1*DIGIT / (1DIGIT '/' 1DIGIT)
Unité         =   'c.s.' / 'c.c.' / 'l' / 'g'
Préparation   =   'Préparation' CRLF 1*Etape
Etape         =   '*' TEXT CRLF
TEXT          =   1*(WSP / VCHAR)
NAME          =   ALPHA *(ALPHA / '-' / DIGIT)
```

3) Spécification du format .srt en ABNF

```
SrtList      =      1*Sous-titre
Sous-titre   =      1*(DIGIT CRLF Timecode WSP '-->' WSP Timecode CRLF Contenu 1*CRLF
Timecode     =      2DIGIT ':' 2DIGIT ':' 2DIGIT ',' 3DIGIT
Contenu      =      1*2(TEXT CRLF)
TEXT         =      1*(WSP / VCHAR)
```